# Course FH3020-1
# NOS Job Control

*Student Handout*

## PROPRIETARY NOTICE

The ideas and designs set forth in this document are the property of Control Data Corporation and are not to be disseminated, distributed, or otherwise conveyed to third persons without the express written permission of Control Data Corporation.

| REVISION RECORD | |
|---|---|
| **REVISION** | **DESCRIPTION** |
| 01 | Manual Release |
| (01-1978) | |
| A | Manual Update |
| (07-31-78) | |
| B | Manual Update |
| (08-31-79) | |
| C | Manual Update |
| (11-30-79) | |
| D | Manual Update |
| (07-15-82) | |
| E | Manual Update to V2.1 |
| (06-15-83) | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| Publication No. FH3020-1 | |

REVISION LETTERS I, O, Q AND X ARE NOT USED

NOS Job Control

Address comments concerning this manual to:

CONTROL DATA CORPORATION
National Coordinator
Bloomington Facility (MNA02B)
5001 West 80th Street
Bloomington, Minnesota 55437

or use Comment Sheet in the back of this manual.

1982
©COPYRIGHT CONTROL DATA CORPORATION 1982
All Rights Reserved

# GENERAL COURSE DESCRIPTION

COURSE TITLE: NOS Job Control

COURSE NUMBER: FH3020

COURSE LENGTH: 5 Days

DESCRIPTION:

This course is intended to prepare a student to effectively use the NOS operating system. The student will learn to set up and run jobs using control language, utilities, permanent files, libraries, loader functions, tapes and UPDATE.

PREREQUISITES:

CYBER 170 NOS Introduction or equivalent plus knowledge of some programming language, preferably FORTRAN or COBOL.

STUDENT MATERIALS:

1. NOS Reference Manual, Volume 1.
2. UPDATE Reference Manual.
3. CYBER Common Loader Reference Manual.
4. NOS Job Control Student Handout.
5. NOS Application Programmers Instant Manual.

# NOS JOB CONTROL
## COURSE CHART

| HOUR | DAY 1 | DAY 2 | DAY 3 | DAY 4 | DAY 5 |
|---|---|---|---|---|---|
| 1 | INTRODUCTION<br>OVERVIEW OF NOS | FILE PRESERVATION AND PERMANENT FILES | SOURCE CODE MAINTENANCE | CYBER CONTROL LANGUAGE | LOADER |
| 2 | GENERAL CONTROL CARDS | | | | |
| 3 | JOB CONTROL LANGUAGE | | LAB | LAB | LAB |
| 4 | FILE STRUCTURE AND UTILITIES | REVIEW | LIBRARY GENERATION | LOADER | TAPES |
| 5 | | | | | |
| 6 | LAB | LAB | LAB | LAB | REVIEW LAB |

COURSE OUTLINE

NOS JOB CONTROL

| | | |
|---|---|---|
| I. | Introduction to NOS | (1 hr.) |
| II. | Job Control Language | (2 hrs.) |
| III. | File Manipulation Utilities | (2 hrs.) |
| IV. | File Preservation & Permanent Files | (3 hrs.) |
| V. | Source Program Library - Maintenance | (2 hrs.) |
| VI. | Introduction to Library Generation | (2 hrs.) |
| VII. | CYBER Control Language | (2 hrs.) |
| VIII. | Loader | (4 hrs.) |
| IX. | Non-Allocatable Resources - Magnetic Tape | (2 hrs.) |

# LESSON 1
# INTRODUCTION TO NOS

## LESSON PREVIEW

This lesson is intended to provide an overview to NOS and to the hardware/software interface.

## OBJECTIVES

Upon the completion of this lesson, the student will:

- Have a basic understanding of the hardware/software interface.
- Be able to set up a simple job.

## TRAINING AIDS

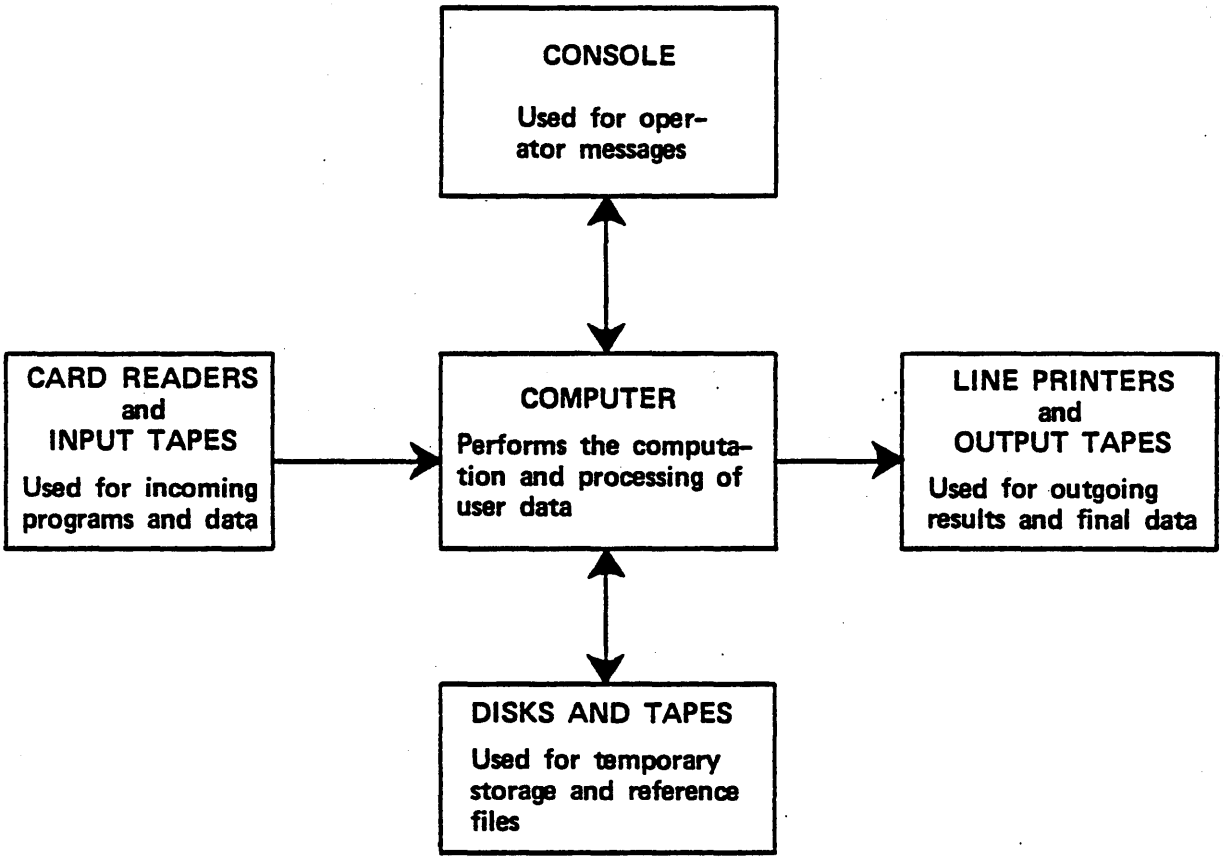
Visuals V1-1 through V1-6.

## PROJECTS

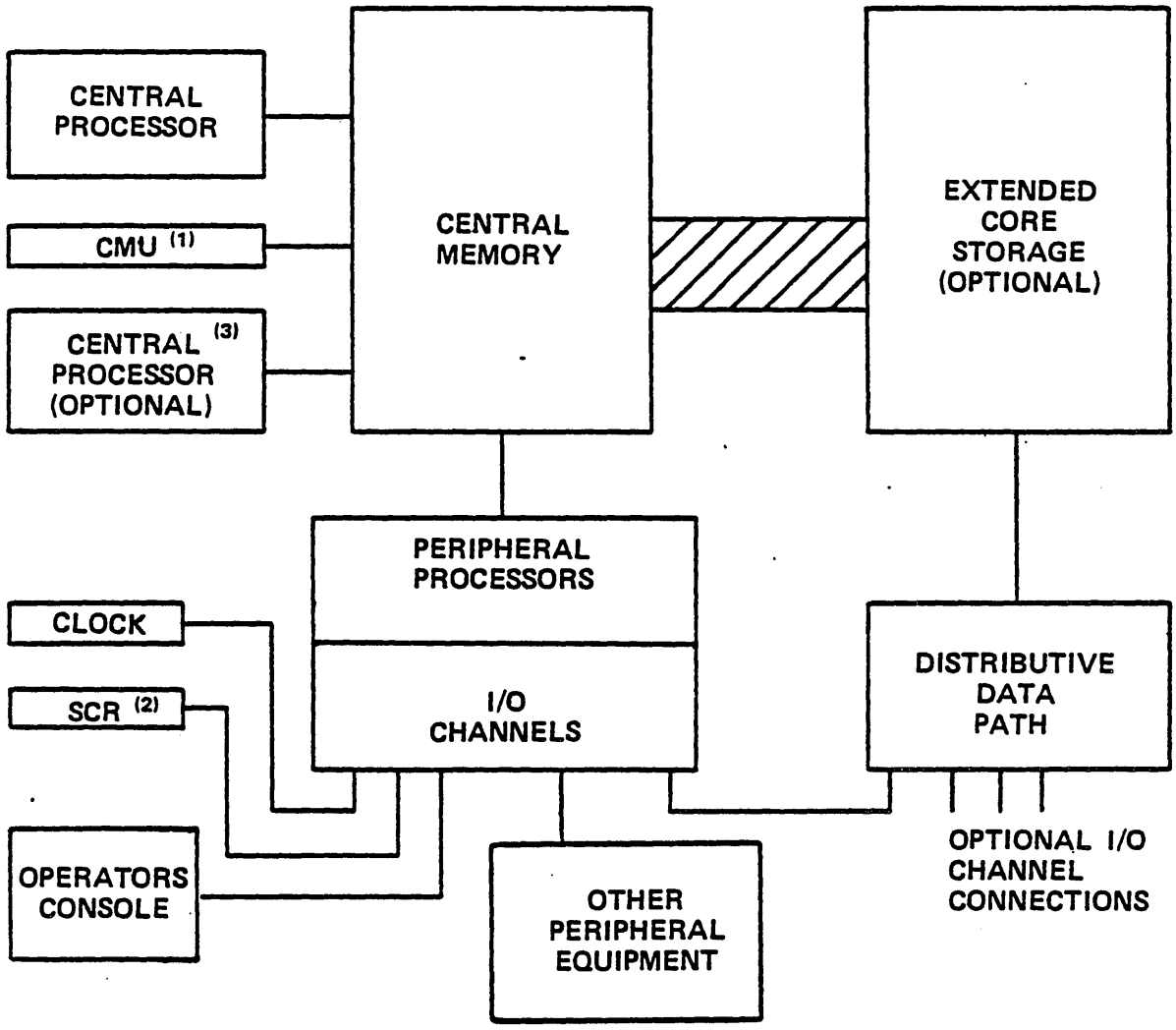
Exercises 1 (problem), 2 (study questions), and 3 (review).

## REFERENCES

NOS Version 1 Reference Manual, Volume 1, Sections 1, 2, 3.

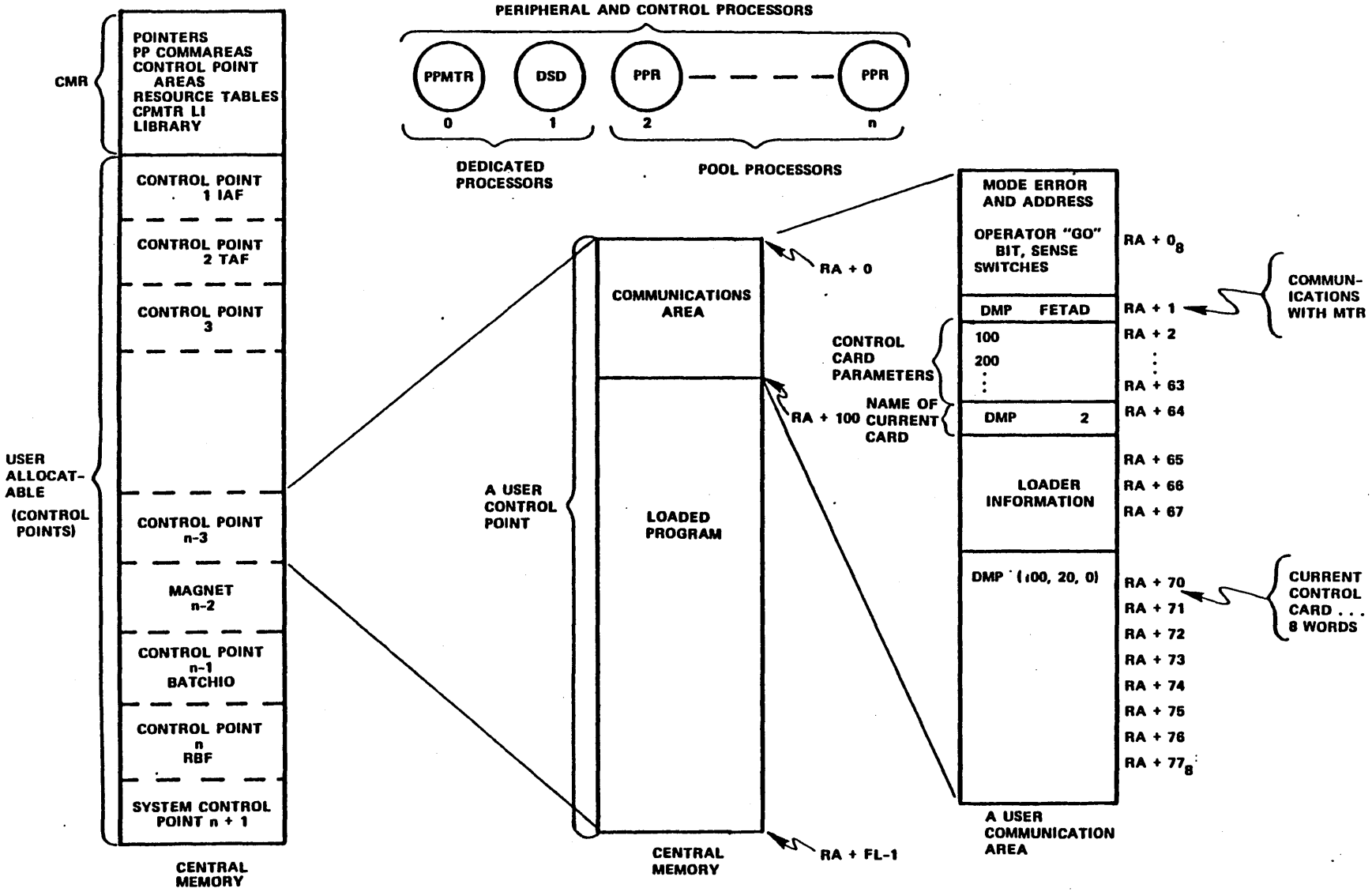

CONSOLE
Used for operator messages
CARD READERS and INPUT TAPES
Used for incoming programs and data
COMPUTER
Performs the computation and processing of user data
LINE PRINTERS and OUTPUT TAPES
Used for outgoing results and final data
DISKS AND TAPES
Used for temporary storage and reference files

Overview of a Computer System




NOTES: 1. Compare/Move Unit (Models 720 and 730 only)
2. Status Control Register
3. Model 720 and 730 only

COMPONENTS OF THE MODELS 720, 730, 750, 760




CYBER 70/NOS Hardware/Software Interface

## JOB FLOW - STAGES

THERE ARE FIVE GENERAL STAGES OR PHASES OF A JOB'S PASSAGE THROUGH THE SYSTEM:

PHASES OF JOB PROCESSING

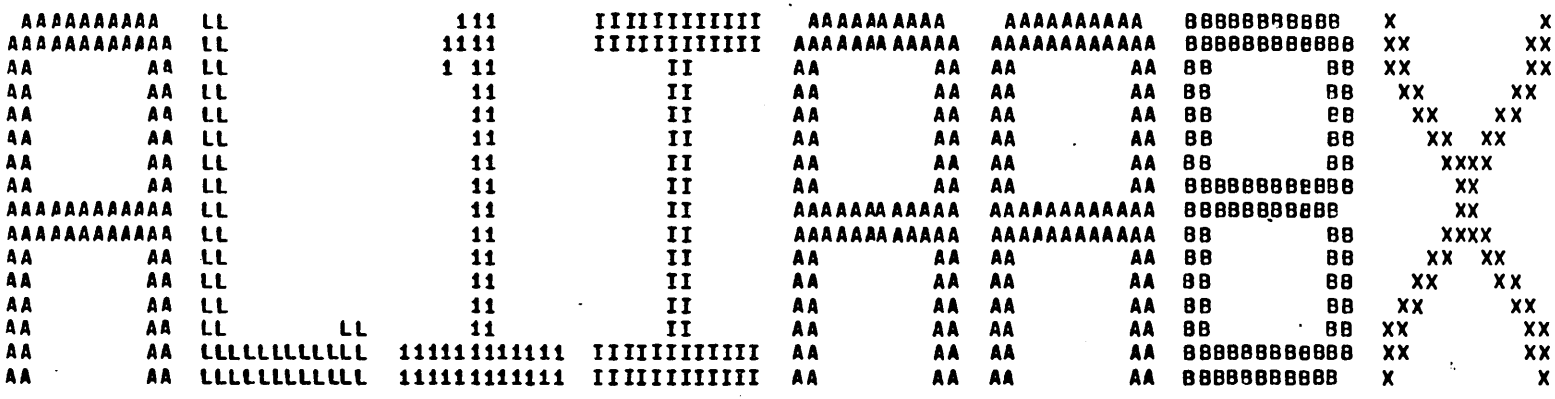
- ENTRY INTO THE SYSTEM
- ASSIGNMENT OF RESOURCES BY THE SYSTEM
- JOB EXECUTION
- TERMINATION OF THE JOB
- DISPOSITION OF THE OUTPUT

```
TEST JOB,T5.
USER,DRF2755,DRF.
FTN.
MAP(PART)
LGO.
R     7/8/9 END-OF-RECORD
       PROGRAM MAIN(OUTPUT)
       PRINT *," IN MAIN"
       CALL SUB1
       END
       SUBROUTINE SUB1
       PRINT *," IN SUB1"
       END
Q     6/7/8/9 END-OF-INFORMATION
```

1

2

```
OPERATING SYSTEM =  NOS 4K30-FNT101      81/10/05. PRINTED = 81/10/07. 11.25.46.

UJN          =  TESTJOB   FAMILY      = SYS172        JOB ORIGIN    = BATCH.
CREATING JSN =  AABX      USER NUMBER = DRF2755       SERVICE CLASS = BATCH.
```



```
      1              PROGRAM MAIN(OUTPUT)
                     PRINT *,* IN MAIN*
                     CALL SUB1
                     END
```

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
2063 MAIN

| FILE NAMES | | MODE |
|---|---|---|
| 0 | OUTPUT | FREE |

| EXTERNALS | TYPE | ARGS |
|---|---|---|
| SUB1 | | 0 |

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 718 | 57 |
| BUFFER LENGTH | 20048 | 1028 |

520008 CM USED

```
      1              SUBROUTINE SUB1
                     PRINT *,* IN SUB1*
                     END
```

SYMBOLIC REFERENCE MAP (R=1)

```
ENTRY POINTS
    1  SUB1

FILE NAMES        MODE
      OUTPUT     FREE

STATISTICS
  PROGRAM LENGTH                  128       10
        52000B CM USED
```

2

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 6657

TRANSFER ADDRESS -- MAIN 2174

PROGRAM ENTRY POINTS -- MAIN 2174

PROGRAM AND BLOCK ASSIGNMENTS:

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| MAIN | 111 | 2075 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | PROGRAM OPT=1 |
| SUB1 | 2206 | 12 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SYSAIC= | 2220 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STF.END/ | 2221 | 1 | | | | | | | |
| /FCL.C./ | 2222 | 30 | | | | | | | |
| /08.IO./ | 2252 | 130 | | | | | | | |
| Q2NTRY= | 2402 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2402 | 43 | | | | | | | |
| COMIO= | 2445 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2500 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL CAPSULE LOADING |
| FECMSK= | 2540 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2601 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2604 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON FLOATING OUTPUT CODE |
| FORSYS= | 3115 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 3616 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL MISC. UTILITIES. |
| GETFIT= | 3643 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| LDOUT= | 3737 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 4203 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON OUTPUT CODE |
| OUTF= | 4357 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 4563 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 4570 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 4630 | 14 | | | | | | | |
| FDL.RES | 4644 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 5055 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 5277 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | 5734 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| ERR$RM | 6013 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 6040 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 6127 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 6134 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 6316 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 6324 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 6340 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 6376 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 6420 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | 6427 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 6635 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

1-10

.537 CP SECONDS 24200B CM STORAGE USED 3 TABLE MOVES

IN MAIN
IN SUB1

```
    FULL  DAYFILE.  81/10/07. 11.25.34.*11.25.22* PAGE    1

11.25.22.TESTJOB,T5.
11.25.22.UCCR, 72070,       0.012KCDS.
11.25.22.USER,ORF2755,.
11.25.23.FTN.
11.25.25.       .094 CP SECONDS COMPILATION TIME
11.25.25.MAP(PART)
11.25.26.LGO.
11.25.32.      END  MAIN
11.25.32.     013100 MAXIMUM EXECUTION FL.
11.25.32.      0.056 CP SECONDS EXECUTICN TIME.
11.25.33.UEAD,       0.001KUNS.
11.25.33.UEMS,       1.873KUNS.
11.25.33.UECF,       0.707SECS.
11.25.33.AESR,       2.176UNTS.
11.25.33.$OUT(*/OP=E)
11.25.33. NO FILES PROCESSED.
11.25.34.$DAYFILE(OUTPUT,JT=D)
11.25.56.UCLP, 72,073,       0.320KLNS.
```

# EXERCISE 1

PROBLEM

GIVEN: Use the job deck as shown below as a basic foundation and perform the following steps on it.

1. (Required) List the job deck using the off-line listing function of the remote job entry station.

2. (Required) Make additions and/or changes to the given sample job so that MAIN also calls SUB2 and SUB3 which produce messages similar to SUB1's output, then make up a complete job deck, submit it for execution and construct a memory layout of the control point from the loader map.

3. Print the contents of the file INPUT.

```
MY JOB,T10.
USER,USERNAME,PASSWORD,FAMILY.
FTN.                                        OFF-LINE
MAP(PART)                                       LISTING
LGO.
00000000000000000000000 EOR
      PROGRAM MAIN (OUTPUT)
      PRINT 1
1     FORMAT (*  IN MAIN*)
      CALL SUB1
      END
      SUBROUTINE SUB1
1     FORMAT (* IN SUB1*)
      PRINT 1
      END
00000000000000000000000 EOI
```

# EXERCISE 2

STUDY QUESTIONS

1. Name the maximum number of programs that NOS controls simultaneously, in addition to its regular functions as an operating system.

2. Name the report printed by NOS at the termination of each batch job that details processing and accounting information about the job.

3. Place the letter identifying each of the following devices in the blank next to its function.

   a. Peripheral Processor
   b. Central Processor
   c. Central Memory

   _____ Contains jobs being executed.

   _____ Transfers information for input/output equipment.

   _____ Primary arithmetic device.

4. Name the 6-bit machine code format used to represent alphanumeric information internally in the computer.

5. Place the letter identifying each of the following items in the blank next to its function.

   a. Control Point
   b. Control Point Area
   c. Exchange Package

   _____ Area in central memory that enables jobs to communicate with central processor.

   _____ Part of low core that contains job name, length, starting address, and other parameters related to the user program.

   _____ Partition of central memory, allocated by NOS, where a job resides while in execution.

6. Number the following events in order in which they occur, if the computer is to execute a job and then print its output.

   _____ The output goes through a peripheral processor printing.

   _____ The job is read from a card reader through two peripheral processors and onto disk storage as part of the input queue.

   _____ The job output on disk is designated part of the output queue.

   _____ The job is assigned a control point number and central memory storage.

   _____ NOS determines the job name, central memory space, and other characteristics needed for the control point area.

_____ NOS determines the job name, central memory space, and other characteristics needed for the control point area.

_____ The job alternately shares use of the central processor with up to twenty two other programs.

7. _____ Place a check mark next to the term used to indicate NOS's automatic arrangement of jobs in central memory to allow enough contiguous storage for an incoming job.

   _____ Storage Move

   _____ Reblocking

   _____ Shifting

   _____ Dynamic Management

8. What are the five ways in which a job may enter the system?

9. What is the first card of a job deck and what is its purpose?

10. (True/False) An EOR card must always precede an EOI card.

11. What is the relationship of the local file, "INPUT" and job deck?

12. Name the stages in job flow in the NOS system.

13. What is the maximum number of jobs that can be in execution simultaneously and the maximum number that can be in some state of execution?

14. List all of the data about a job that is both reported and retained by the system.

15. Discuss what happens when a fatal error occurs.

16. How does the scheduler keep track of the different types of jobs in the system?

# EXERCISE 3

REVIEW:

Define the following:

1) Central Processor
2) Central Memory
3) Extended Core Storage
4) Peripheral Processor
5) Channel
6) Peripheral Device
7) Distributive Data Path
8) Compare Move Unit
9) Central Memory Resident
10) User Area
11) Control Point
12) Communication Area
13) Reference Address
14) Field Length

# LESSON 2
# JOB CONTROL LANGUAGE

## LESSON PREVIEW

In this lesson, the student will learn how to structure a job by the use of job control statements. This includes the identification of a job and the specification of needed resources by means of the following control statement:

| | | | | |
|---|---|---|---|---|
| USER | CHARGE | COMMENT | CTTIME | DAYFILE |
| EXIT | LIMITS | MODE | NOEXIT | SWITCH |
| JOBCARD | PASSWOR | RTIME | SETTL | |

## OBJECTIVES

Upon completion, the student will be able to:

- State the rules of syntax governing formation of names used by the system and of job control statements.
- Use the USER and CHARGE control statements to gain access to the system.
- Use the LIMITS control statement to receive a report of allowable resources for an account number.
- Use the PASSWOR control statement to change the password on the USER control statement.
- Use the DAYFILE, COMMENT, *, and EXIT control statements.
- List the contents of the user's communication area in locations RA+0 through RA+100.
- Use the MODE statement and judge the consequences of its use.
- Use the SETTL statement to change CPU time limit.

## TRAINING AIDS

Visuals V2-1 through V2-12.

## PROJECTS

Exercises 4 (study questions) and 5 (problem).

## REFERENCES

NOS Version 1 Reference Manual, Volume 1, Sections 5, 6.

## NOS JOB CONTROL AND ACCOUNTING

Before looking at the Job Control control cards, it is important to look at the NOS User/Accounting structure.

At system installation time, some or all of the mass storage devices are designated as Master Devices.

The Master Devices hold all permanent file catalog information, for both direct access and indirect access files.

The Master Devices also contain all indirect access files completely and may also be designated at installation of CMRDECK to contain direct access files also or not.

The primitive element of the NOS accounting structure is a simple octal number, ranging from $0_8$ to $377777_8$, called a User Index.

Each User Index is mapped on to a particular Master Device by the unit digit of the User Index number, to allow distribution of permanent files to be appropriately even among Master Devices.

The User Index is the key, unique element in the NOS accounting system and has one or more Account Numbers associated with it by means of the use of the MODVAL program run by the central site systems analyst.

The User Name is the billable entity that is required for access to a NOS system that has the accounting functions enabled, and the accounting procedure for system resources used, though tailored to local operating policies and realities, is nevertheless the single identifier for a billable account.

```
JOBCRD1,T2.
USER(QUIK172,QUIK)
LIMITS.
* THIS IS A COMMENT CARD DEFINED BY AN ASTERISK IN COLUMN ONE.
COMMENT. THIS IS ALSO A COMMENT CARD
   COMMENT. A CONTROL STATEMENT NEED NOT START IN COLUMN ONE.
DAYFILE.
COMMENT.THERE ISNT ANY CONTINUATION FOR CONTROL STATEMENTS EXCEPT THOSE LISTED
IN SECTION 10 OF THE N.O.S. REF.MAN.
EXIT. THIS CARD REQUIRED TO ALLOW JOB TO CONTINUE.
PASSWOR,QUIK,SLOW. CHANGE PASSWORD
USER(ADV4077,ADV4)
EXIT. WE BETTER CHANGE OUR PASSWORD BACK
PASSWOR,SLOW,QUIK.
EXIT.
Q     6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/19. 19.09.41.*19.09.39* PAGE    1

19.09.39.JOBCRD1,T2.
19.09.39.UCCR, 90020,       0.015KCDS.
19.09.39.USER(QUIK172,)
19.09.40.LIMITS.                    1
19.09.40. INQUIRY COMPLETE.
19.09.40.* THIS IS A COMMENT CARD DEFINED BY AN ASTERISK IN
19.09.40. COLUMN ONE.
19.09.40.COMMENT. THIS IS ALSO A COMMENT CARD
19.09.40.   COMMENT. A CONTROL STATEMENT NEED NOT START IN COLUMN ONE.
19.09.40.DAYFILE.
19.09.40. USER DAYFILE PROCESSED.   2
19.09.40.COMMENT.THERE ISNT ANY CONTINUATION FOR CONTROL STATEMENTS EXCEPT THOSE LISTED
19.09.40.IN SECTION 10 OF THE N.O.S. REF.MAN.
19.09.40. FORMAT ERROR ON CONTROL CARD.
19.09.40.EXIT. THIS CARD REQUIRED TO ALLOW JOB TO CONTINUE.
19.09.40.PASSWOR,,.          CHANGE PASSWORD
19.09.41. UPDATE COMPLETE.
19.09.41. ILLEGAL USER CARD.
19.09.41. SYSTEM ABORT.
19.09.41.UEAD,       0.001KUNS.
19.09.41.UEPF,       0.008KUNS.
19.09.41.UEMS,       0.582KUNS.
19.09.41.UECP,       0.053SECS.
19.09.41.AESR,       1.143UNTS.
19.09.41.$OUT(*/OP=E)
19.09.41. NO FILES PROCESSED.
19.09.41.$DAYFILE(OUTPUT,JT=O)
```

LIMITS FOR QUIK172 FM/FEATURE 81/10/19. 19.09.40.

MAXIMUM NUMBER OF -
 MAGNETIC TAPE UNITS THAT MAY BE ASSIGNED..... 2
 REMOVABLE AUXILIARY DEVICES THAT MAY BE ASSIGNED 2
 CPU SECONDS ALLOWED FOR EACH JOB STEP/10B..... UNLIMITED
 CENTRAL MEMORY WORDS ALLOWED/100B..... UNLIMITED
 EXTENDED MEMORY WORDS ALLOWED/10[illegible]B..... 0B
 JOBS THAT CAN BE DETACHED..... 1
 DEFERRED BATCH FILES..... UNLIMITED
 PERMANENT FILES ALLOWED IN YOUR CATALOG..... UNLIMITED
 PRUS AVAILABLE FOR ANY ONE INDIRECT FILE..... UNLIMITED
 PRUS AVAILABLE FOR ANY ONE DIRECT FILE..... UNLIMITED
 PRUS AVAILABLE FOR ALL INDIRECT FILES..... UNLIMITED
 PRUS AVAILABLE PER JOB..... UNLIMITED
 MESSAGES THAT MAY BE ISSUED TO THE DAYFILES..... UNLIMITED
 BATCH CONTROL STATEMENTS THAT MAY BE PROCESSED.. UNLIMITED
 CARDS THAT MAY BE PUNCHED PER JOB..... UNLIMITED
 LINES THAT MAY BE PLOTTED FOR PLOT FILE..... 512
 LINES THAT MAY BE PRINTED PER JOB..... UNLIMITED
 SRUS ALLOWED PER JOB..... UNLIMITED
 SYSTEM PROCEDURE FILE THAT IS EXECUTED.....PROC!!

TERMINAL CHARACTERISTICS -
 INITIAL SUBSYSTEM..... NULL
 INITIAL CHARACTER SET TO BE USED..... STANDARD
 TERMINAL PARITY..... EVEN
 TRANSMISSION MODE..... HALF
 TERMINAL TYPE..... TTY

THE FOLLOWING ARE VALID USER PERMISSIONS -
 CHANGE YOUR PASSWORD
 USE PRIVILEGED TIME-SHARING COMMANDS
 CREATE DIRECT ACCESS FILES
 CREATE INDIRECT ACCESS FILES
 HAVE SYSTEM ORIGIN PRIVILEGES
 ACCESS/CREATE LIBRARY FILES
 ASSIGN MAGNETIC TAPES
 ACCESS SYSTEM WITHOUT SUPPLYING CHARGE/PROJECT NUMBER
 ACCESS OR CREATE FILES ON AUXILIARY DEVICES
 ACCESS TRANSACTION FUNCTIONS FOR LIBRARY UPDATES
 HAVE UNLIMITED TERMINAL CONNECT TIME WITHOUT TIMEOUT
 USE SYSTEM CONTROL POINT FACILITY
 HAVE SPECIAL ACCOUNTING PRIVILEGES
 USE BATCHIO SUBSYSTEM PRIVILEGES
 USE THE PROTECT STATEMENT

THE FOLLOWING ARE VALID NETWORK APPLICATIONS -
 INTERACTIVE FACILITY (IAF)
 REMOTE BATCH FACILITY (RBF)
 TRANSACTION FACILITY (TAF)
 MESSAGE CONTROL SYSTEM (MCS)
 TERMINAL VALIDATION FACILITY (TVF)

1

2-4

2

FULL DAYFILE. 81/10/19. 19.09.40.*19.09.39* PAGE 1

19.09.39.JOBCRD1,T2.
19.09.39.UCCR, 9JJ2J, 0.015KCDS.
19.09.39.USER(QUIK172,)
19.09.40.LIMITS.
19.09.40. INQUIRY COMPLETE.
19.09.40.* THIS IS A COMMENT CARD DEFINED BY AN ASTERISK IN
19.09.40. COLUMN ONE.
19.09.40.COMMENT. THIS IS ALSO A COMMENT CARD
19.09.40. COMMENT. A CONTROL STATEMENT NEED NOT START IN COLUMN ONE.
19.09.40.DAYFILE.

```
JOBCRD2,T2.
USER,QUIK172,SLOW.
PASSWOR,SLOW,QUIK.            CHANGE PASSWORD BACK
Q     6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/19. 19.10.28.♦19.10.27♦ PAGE     1

19.10.27.JOBCRD2,T2.
19.10.27.UCCR, 90020,       0.003KCDS.
19.10.27.USER,QUIK172,.
19.10.27.PASSWOR,,,.                        CHANGE PASSWORD BACK
19.10.27. UPDATE COMPLETE.
19.10.27.UEAD,        0.001KUNS.
19.10.27.UEPF,        0.004KUNS.
19.10.27.UEMS,        0.333KUNS.
19.10.27.UECP,        0.021SECS.
19.10.27.AESR,        1.374UNTS.
19.10.27.$OUT(*/OP=E)
19.10.27. NO FILES PROCESSED.
19.10.28.$DAYFILE(OUTPUT,JT=D)
```

The sense switches are located in bits 11-6 of control point location RA+0.

Relocatable object programs avoid the user communication area and are laid out from RA+111B to RA+FL-1.

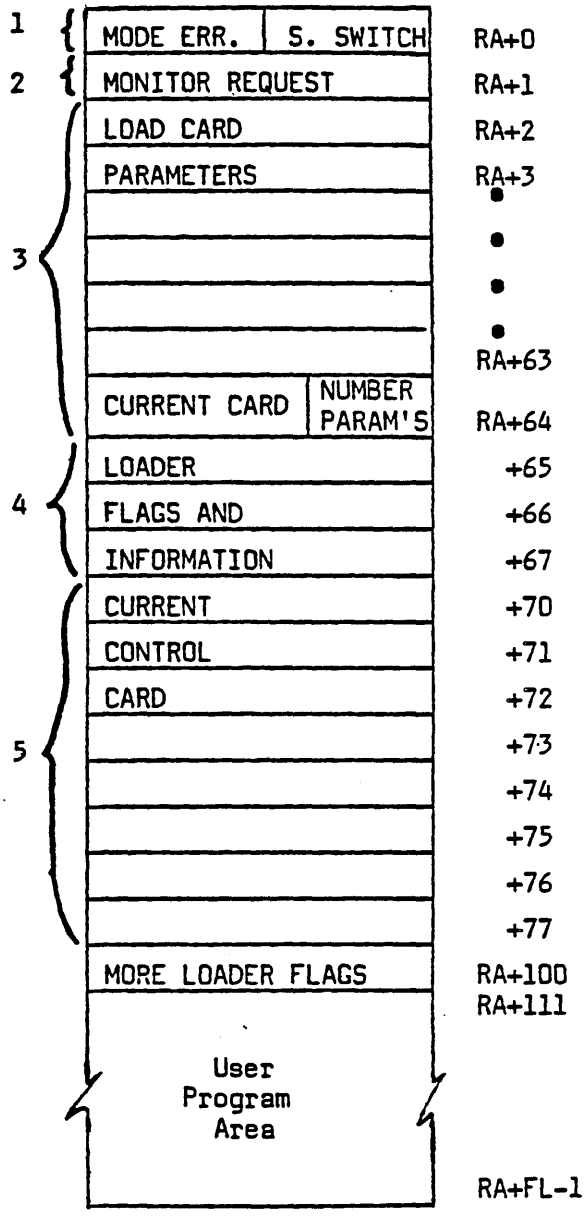
Each word in the are RA+0 to RA+77 is assigned a specific function by design.

1. MODE and SWITCH affect RA+0

2. System and File Action requests to monitor are posted here

3. Are the current control card
and 5. in two different formats

4. Type of load




A User Communication Area

```
TMLIMIT,T50.
USER,DRF2755,DRF.
FTN,R=0,B=ENDLESS.
SETTL(4)
ENDLESS.
EXIT.
CTIME.
RTIME.
SETTL(60)
RETURN,ENDLESS.
COMPASS(R=ENDLESS,LO=RR)
ENDLESS.
EXIT.
CTIME.
RTIME.
R
      PROGRAM TEST(OUTPUT)
    1 I=1
      IF(I.NE.0) GO TO 1
      END
R    7/8/9 END-OF-RECORD
           IDENT  LOOP
           ENTRY  START
 START     NO
+          NO
+          EQ     START
           END    START
R
Q    6/7/8/9 END-OF-INFORMATION
```

1

2

```
1                PROGRAM TEST(OUTPUT)
               1 I=1
                 IF(I.NE.0) GO TO 1
                 END
```

1

2

```
                                    IDENT  LOOP
                                    ENTRY  START
 0  46000                  START    NO
 1  46000                  *        NO
 2  0400000000 *           *        EQ     START
 3                                   END    START

          437008 CM  STORAGE USED              6 STATEMENTS        1 SYMBOLS
                        SERIAL CPU ASSEMBLY          0.039 SECONDS          0 REFERENCES
```

```
    FULL  DAYFILE.  81/10/07. 11.16.28.*11.15.11* PAGE    1

11.15.11.TMLIMIT,T50.
11.15.11.UCCR, 72070,      0.025KCDS.
11.15.11.USER,DRF2755,.
11.15.12.FTN,R=0,B=FNDLESS.
11.15.13.      .052 CP SECONDS COMPILATION TIME
11.15.13.SETTL(4)
11.15.13.ENDLESS.
11.15.19. TIME LIMIT.
11.15.19.EXIT.
11.15.19.CTIME      4.555 SECS.
11.15.19.RTIME   2614.047 SECS.
11.15.19.SETTL(60)
11.15.19.RETURN,ENDLESS.
11.15.19.COMPASS(B=ENDLESS,LO=BR)
11.15.20. ASSEMBLY COMPLETE.   437008  CM USED.
11.15.20.    0.241  CPU SECONDS  ASSEMBLY TIME.
11.15.20.ENDLESS.
11.16.28. TIME LIMIT.
11.16.28.EXIT.
11.16.28.CTIME     66.067 SECS.
11.16.28.RTIME   2683.122 SECS.
11.16.28.UEAD,      0.001KUNS.
11.16.28.UEMS,      1.533KUNS.
11.16.28.UECP,     66.067SECS.
11.16.28.AESR,     67.738UNTS.
11.16.28.$OUT(*/OP=E)
11.16.28. NO FILES PROCESSED.
11.16.28.$DAYFILE(OUTPUT,JT=0)
11.16.40.UCLP, 72,073,      0.256KLNS.
```

```
MODERR,T4.
USER,DRF2755,DRF.
FTN.                    1
MAP(PART)
LGO.
EXIT.  2
*
* TRY AGAIN WITH ERROR SUPPRESSED
*
MODE,5.
LGO.
DMP.
R     7/8/9 END-OF-RECORD
       PROGRAM BUGS(OUTPUT)
       NAMELIST /X/ A,B,C,D
       B=1.E322
       A=B
       C=A+B
       D=1
       PRINT X
*
*      THE FOLLOWING STATEMENT WILL CAUSE AN ERROR CONDITION
*
       D=C+D
       PRINT X
       PRINT *," ERROR DID NOT ZAP US"
       END
Q     6/7/8/9 END-OF-INFORMATION
```

```
      1              PROGRAM BUGS(OUTPUT)
                     NAMELIST /X/ A,B,C,D
                     B=1.E322
                     A=B
      5              C=A+B
                     D=1
                     PRINT X
                *
                *    THE FOLLOWING STATEMENT WILL CAUSE AN ERROR CONDITION
     10         *
                     D=C+D
                     PRINT X
                     PRINT *,# ERROR DID NOT ZAP US#
                     END
```

1

SYMBOLIC REFERENCE MAP (R=1)

ENTRY POINTS
 2062  BUGS

| VARIABLES | SN | TYPE | RELOCATION | | | | |
|---|---|---|---|---|---|---|---|
| 2124 | A | REAL | | 2125 | B | REAL | |
| 2126 | C | REAL | | 2127 | D | REAL | |

| FILE NAMES | | MODE |
|---|---|---|
| 0 | OUTPUT | MIXED |

NAMELISTS
 X

STATISTICS

| | | |
|---|---|---|
| PROGRAM LENGTH | 1318 | 89 |
| BUFFER LENGTH | 20038 | 1027 |

520008 CM USED

2-11

**2**

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 6764

TRANSFER ADDRESS -- BUGS 2173

PROGRAM ENTRY POINTS -- BUGS 2173

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| BUGS | 111 | 2134 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | PROGRAM OPT=1 |
| SYSAID= | 2245 | 1 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2246 | 1 | | | | | | | |
| /FCL.C./ | 2247 | 27 | | | | | | | |
| /Q8.IO./ | 2276 | 127 | | | | | | | |
| Q2NTRY= | 2425 | 0 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2425 | 41 | | | | | | | |
| COMIO= | 2466 | 33 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2521 | 40 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL CAPSULE LOADING |
| FECMSK= | 2561 | 41 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2622 | 3 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2625 | 311 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON FLOATING OUTPUT CODE |
| FORSYS= | 3136 | 376 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 3534 | 25 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL MISC. UTILITIES. |
| GETFIT= | 3561 | 74 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| LDOUT= | 3655 | 240 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LIST DIRECTED OUTPUT FORMATTING |
| NAMOUT= | 4115 | 264 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | NAMELIST OUTPUT ROUTINE. |
| OUTCOM= | 4401 | 154 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON OUTPUT CODE |
| OUTF= | 4555 | 172 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 4747 | 5 | SL-SYSLIB | 81/09/18 | COMPASS | 3.6 | 552 | | 79/05/10. 81/08/07. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 4754 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 5014 | 14 | | | | | | | |
| FDL.RES | 5030 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 5241 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 5463 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| ERR$RM | 6120 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 6145 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 6234 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 6241 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 6423 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 6431 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 6445 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 6503 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 6525 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | 6534 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 6742 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

.312 CP SECONDS 242008 CM STORAGE USED 3 TABLE MOVES

```
$X

A       = .1+323,

B       = .1+323,

C       =   R,

D       = .1E+01,

$END
```

```
   EXCHANGE PACKAGE.          VERSION = NOS 4F-FNT100          81/10/01. 11.29.13.
   P          0  A0       0  B0       0   (A0) 0002 0022 0400 0000 0000
   RA    107400  A1    2226  B1       1   (A1) 0000 0000 0000 0000 0111
   FL     11000  A2     124  B2    1566   (A2) 0000 0005 6203 2000 0000
   EM      7007  A3     173  B3     174   (A3) 3353 0516 0400 0000 0000
   RAE        0  A4    2237  B4      12   (A4) 3777 6245 7211 5430 5540
   FLE        0  A5    2240  B5      12   (A5) 1720 4000 0000 0000 0000
   MA       400  A6    2276  B6    4316   (A6) 3353 0516 0455 5555 5555
                 A7     147  B7      52   (A7) 0104 0001 7400 0001 4400

   X0   3777 0000 0000 0000 0000
   X1   0000 0000 0000 0000 0111
   X2   7777 7777 7777 7777 7776
   X3   3353 0516 0400 0000 0000
   X4   3777 6245 7211 5430 5540
   X5   1720 4000 0000 0000 0000
   X6   0000 0000 0000 0000 0142
   X7   0104 0001 7400 0001 4400

(RA)    0002 0022 0400 0000 0000
(RA+1)  0000 0000 0000 0000 0000
```

```
CM DUMP FROM    2144 TO    2244.      VERSION = NOS 4F-FNT100       81/10/01. 11.29.13.
  2144      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
          DUPLICATED LINES.
  2164      00000 00000 00000 00000      20010 00000 00000 02170      00000 00000 00000 11610      00000 00000 00000 00000
  2170      17252 42025 24000 00111      00000 00000 00000 00000      02250 72355 55550 02173      04000 02174 00000 00001
  2174      51500 02234 71000 00001      51100 02223 27700 30655      24707 51600 02237 10655      51700 02240 51600 02236
  2200      51600 02235 46000 46000      01000 04123 00070 02172      51500 02240 51400 02237      51100 02226 30054 24700
  2204      54750 46000 46000 46000      01000 04123 00140 02172      51100 02231 46000 46000      01000 04571 00150 02172
  2210      51100 02172 04000 03217      30000 00000 00000 00000      01000 00000 00000 00003      00000 00000 00000 02235
  2214      02000 00000 00000 00003      00000 00000 00000 02236      03000 00000 00000 00003      00000 00000 00000 02237
  2220      04000 00000 00000 00003      00000 00000 00000 02240      00000 00000 00000 00000      00000 00000 00000 00111
  2224      00000 00000 00000 02211      00000 00000 00000 00000      00000 00000 00000 00111      00000 00000 00000 02211
  2230      00000 00000 00000 00000      00000 00000 00000 00111      00060 00000 25000 02241      00000 00000 00000 00000
  2234      37766 24572 11543 05540      37766 24572 11543 05540      37766 24572 11543 05540      37776 24572 11543 05540
  2240      17204 00000 00000 00000      55052 22217 22550 41104      55161 72455 32012 05525      23555 55555 55555 55555
  2244      00000 00000 00000 00000      01000 03364 00000 00000      04000 03273 61000 46000      55555 55555 55555 55555
```

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 6764

TRANSFER ADDRESS -- BUGS 2173

PROGRAM ENTRY POINTS -- BUGS 2173

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| BUGS | 111 | 2134 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | PROGRAM OPT=1 |
| SYSAID= | 2245 | 1 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2246 | 1 | | | | | | | |
| /FCL.C./ | 2247 | 27 | | | | | | | |
| /OR.IO./ | 2276 | 127 | | | | | | | |
| Q2NTRY= | 2425 | 0 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2425 | 41 | | | | | | | |
| COMIO= | 2466 | 33 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2521 | 40 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL CAPSULE LOADING |
| FECMSK= | 2561 | 41 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2622 | 3 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2625 | 311 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON FLOATING OUTPUT CODE |
| FORSYS= | 3136 | 376 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 3534 | 25 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL MISC. UTILITIES. |
| GETFIT= | 3561 | 74 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| LDOUT= | 3655 | 240 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LIST DIRECTED OUTPUT FORMATTING |
| NAMOUT= | 4115 | 264 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | NAMELIST OUTPUT ROUTINE. |
| OUTCON= | 4401 | 154 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON OUTPUT CODE |
| OUTF= | 4555 | 172 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 4747 | 5 | SL-SYSLIB | 81/09/18 | COMPASS | 3.6 | 552 | | 79/05/10. 81/08/07. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 4754 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 5014 | 14 | | | | | | | |
| FDL.RES | 5030 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 5241 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 5463 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| ERR$RM | 6120 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 6145 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 6234 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 6241 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 6423 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 6431 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 6445 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 6503 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 6525 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | 6534 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 6742 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

.323 CP SECONDS 242008 CM STORAGE USED 3 TABLE MOVES

```
A       = .1+323,
B       = .1+323,
C       =   R,
D       = .1E+01,
$END
```

```
$X
A       = .1+323,
B       = .1+323,
C       =   R,
D       =   R,
$END
 ERROR DID NOT ZAP US
```

```
EXCHANGE PACKAGE.          VERSION = NOS 4F-FNT100          81/10/01. 11.29.16.
P       4756   A0     111   B0       0   (A0) 1725 2420 2524 0000 0131
RA    107400   A1       1   B1       1   (A1) 0516 0420 0000 0000 0000
FL     11200   A2   11137   B2  777755   (A2) 0000 0000 0000 0000 0111
EM      7005   A3    3156   B3  771013   (A3) 5555 5555 0000 0000 0000
RAE        0   A4    3137   B4      33   (A4) 2000 0000 0000 0000 1444
FLE        0   A5    2270   B5       0   (A5) 2000 0000 0000 0000 1376
MA       400   A6       1   B6  775777   (A6) 0516 0420 0000 0000 0000
               A7    6523   B7      30   (A7) 0000 0000 0000 0000 0001

X0   7777 7777 7777 7777 7770
X1   0000 0000 0000 0000 0000
X2   0000 0000 0000 0000 0111
X3   5555 5555 0000 0000 0000
X4   0000 0000 0000 0000 0000
X5   0000 0000 0000 0000 0033
X6   0516 0420 0000 0000 0000
X7   0000 0000 0000 0000 0001

(RA)   0000 0000 0000 0000 0000
(RA+1) 0516 0420 0000 0000 0000


CM DUMP FROM    4716 TO    5016.      VERSION = NOS 4F-FNT100          81/10/01. 11.29.16.
  4714      61247 77765 04200 04716      10766 54761 61000 46000      51100 02317 51300 02340      03130 04736 52510 00021
  4720      10355 20350 03330 04736      51200 02257 52510 00022      43736 15357 37332 46000      03230 04707 51300 02342
  4724      03030 04736 51100 02317      52510 00021 10255 20250      03220 04736 66611 67776      07700 04736 73546 46000
  4730      71705 50013 76240 20760      61500 04734 20273 46000      03220 04461 61660 00001      20766 73546 04000 04461
  4734      72450 00000 61200 00012      04420 04736 50660 00001      53010 53010 10344 66500      61600 04741 61000 46000
  4740      04000 06053 42000 00000      51300 02343 61400 00011      27743 13666 53230 46000      50600 00047 51700 02330
  4744      10622 53630 71600 00055      20666 04000 04713 46000      04000 03700 61000 46000      01000 04756 61000 46000
  4750      04000 10035 00000 00000      43644 15116 20230 46000      71601 50065 12121 20650      12661 04000 04747 46000
  4754      04000 04767 00000 00000      01300 00000 00000 00000      04000 03235 00000 00000      51100 00001 03110 04757
  4760      54610 04000 04755 46000      51100 00066 03310 04763      51100 04754 04000 04764      71100 00130 20160 46000
  4764      13661 13161 13661 46000      51600 04755 10611 46000      51100 00001 01000 04754      20652 01000 04756 46000
  4770      51100 00001 03110 04770      04004 04771 61000 46000      51100 00001 03110 04771      71602 20314 04000 04767
  4774      20150 36661 01000 04756      04004 04775 61000 46000      71602 20314 20652 36662      53160 20173 03310 04775
  5000      03010 04775 51100 00001      03110 04777 71100 00001      04000 04774 61000 46000      71603 24616 12661 20651
  5004      01000 04756 61000 46000      04000 03216 00000 00000      73660 20630 12161 73610      20123 03210 05003 20151
  5010      13116 20636 51600 05013      74660 36116 20123 46000      04000 05003 61000 46000      00000 03144 00000 00001
  5014      00000 00000 00000 06766      00000 00000 00000 06767      00000 00000 00000 05263      00000 00000 00000 00111
```

```
    FULL  DAYFILE.  81/10/01. 11.29.17.*11.29.10* PAGE    1

11.29.10.MODERR,T4.
11.29.10.UCCR, 64020,      0.026KCDS.
11.29.10.USER,DRF2755,,.
11.29.10.FTN.
11.29.11.      .076 CP SECONDS COMPILATION TIME
11.29.11.MAP(PART)
11.29.11.LGO.
11.29.13. CPU ERROR EXIT AT  002204.
11.29.13. ARITHMETIC OVERFLOW.
11.29.14.EXIT.
11.29.14.*
11.29.14.* TRY AGAIN WITH ERROR SUPPRESSED
11.29.14.*
11.29.14.MODE,5.
11.29.14.LGO.
11.29.16.      END  BUGS
11.29.16.    013200 MAXIMUM EXECUTION FL.
11.29.16.      .038 CP SECONDS EXECUTION TIME.
11.29.16.DMP.
11.29.17.UEAD,       0.001KUNS.
11.29.17.UEMS,       2.197KUNS.
11.29.17.UECP,       0.817SECS.
11.29.17.AESR,       2.322UNTS.
11.29.17.$OUT(*/OP=E)
11.29.17. NO FILES PROCESSED.
11.29.17.$DAYFILE(OUTPUT,JT=D)
11.34.01.UCLP, 64,023,      0.640KLNS.
```

```
ERR1.
USER,DRF2755,DRF.
BADCOMMAND.
OTHERBAD.
EXIT. IF ALL GOES WELL, JOB ENDS HERE
* IN CASE OF ERROR CONTINUE HERE
* COULD TAKE A DUMP
* OR MAYBE SAVE SOME FILES
Q     6/7/8/9 END-OF-INFORMATION
```

```
     FULL  DAYFILE.  81/10/01. 11.29.06.*11.29.05* PAGE     1

11.29.05.ERR1.
11.29.05.UCCR, 64020,       0.008KCDS.
11.29.05.USER,DRF2755,.
11.29.05.BADCOMMAND.
11.29.05. FORMAT ERROR ON CONTROL CARD.
11.29.05.EXIT. IF ALL GOES WELL, JOB ENDS HERE
11.29.05.* IN CASE OF ERROR CONTINUE HERE
11.29.05.* COULD TAKE A DUMP
11.29.05.* OR MAYBE SAVE SOME FILES
11.29.05.UEAD,        0.001KUNS.
11.29.05.UEMS,        0.005KUNS.
11.29.05.UECP,        0.001SECS.
11.29.05.AESR,        1.002UNITS.
11.29.05.$OUT(*/OP=E)
11.29.06. NO FILES PROCESSED.
11.29.06.$DAYFILE(OUTPUT,JT=D)
11.33.21.UCLP, 64,023,       0.128KLNS.
```

```
ERR2,T1.
USER,DRF2755,DRF.
NOEXIT.
BADONE.
MOREBAD.   KEEP TRYING
MOSTBAD.   NO MATTER HOW BAD IT GETS
Q     6/7/8/9 END-OF-INFORMATION
```

```
     FULL  DAYFILE.  81/10/01. 11.29.07.*11.29.06* PAGE    1

11.29.06.ERR2,T1.
11.29.06.UCCR, 64020,      0.006KCDS.
11.29.07.USER,DRF2755,.
11.29.07.NOEXIT.
11.29.07.BADONE.
11.29.07. ILLEGAL CONTROL CARD.
11.29.07.MOREBAD.   KEEP TRYING
11.29.07. ILLEGAL CONTROL CARD.
11.29.07.MOSTBAD.   NO MATTER HOW BAD IT GETS
11.29.07. ILLEGAL CONTROL CARD.
11.29.07.UEAD,      0.001KUNS.
11.29.07.UEMS,      0.017KUNS.
11.29.07.UECP,      0.002SECS.
11.29.07.AESR,      1.004UNTS.
11.29.07.$OUT(*/OP=E)
11.29.07. NO FILES PROCESSED.
11.29.07.$DAYFILE(OUTPUT,JT=D)
11.33.28.UCLP, 64,023,      0.128KLNS.
```

```
FTNERR.
USER,DRF2755,DRF.
MAP(PART)
FTN,R=0,GO.
EXIT.
R
      PROGRAM TEST(OUTPUT)
      PRINT *," THIS MESSAGE COMES FROM PROGRAM TEST"
      CALL A
      END
      SUBROUTINE A
      PRINT *," THIS MESSAGE COMES FROM SUBROUTINE A CALLED BY TEST"
      CALL B
      END
      SUBROUTINE B
      PRINT *," THIS MESSAGE COMES FROM SUBROUTINE B CALLED BY A"
      CALL C
      END
      SUBROUTINE C
      PRINT *," THIS MESSAGE COMES FROM SUBROUTINE C CALLED BY B"
      CALL D
      END
      SUBROUTINE D
      PRINT *," THIS MESSAGE COMES FROM SUBROUTINE D CALLED BY C"
      CALL E
      END
      SUBROUTINE E
      PRINT *," THIS MESSAGE COMES FROM SUBROUTINE E CALLED BY D"
      CALL F
      END
      SUBROUTINE F
      PRINT *," THIS MESSAGE COMES FROM SUBROUTINE F CALLED BY E"
C     THIS IS COMMENT CARD NUMBER 1
C     THIS IS COMMENT CARD NUMBER 2
      I=2
      J=4
      K=I+J
C     THIS IS COMMENT CARD NUMBER 3
C     THIS IS COMMENT CARD NUMBER 4
      PRINT *,I,J,K
      CALL ERROR
C     THIS IS COMMENT CARD NUMBER 5
      END
Q
```

1

```
    PROGRAM TEST       73/74   OPT=1                    FTN 4.8+528        81/10/01. 11.29.13        PAGE      1

1                   PROGRAM TEST(OUTPUT)
                    PRINT *,# THIS MESSAGE COMES FROM PROGRAM TEST#
                    CALL A
                    END
```

1

```
 SUBROUTINE A       73/74   OPT=1                    FTN 4.8+528        81/10/01. 11.29.13        PAGE      1

1                   SUBROUTINE A
                    PRINT *,# THIS MESSAGE COMES FROM SUBROUTINE A CALLED BY TEST#
                    CALL B
                    END
```



```
 SUBROUTINE B       73/74   OPT=1                    FTN 4.8+528        81/10/01. 11.29.13        PAGE      1

1                   SUBROUTINE B
                    PRINT *,# THIS MESSAGE COMES FROM SUBROUTINE B CALLED BY A#
                    CALL C
                    END
```

```
 SUBROUTINE C       73/74   OPT=1                    FTN 4.8+528        81/10/01. 11.29.13        PAGE      1

1                   SUBROUTINE C
                    PRINT *,# THIS MESSAGE COMES FROM SUBROUTINE C CALLED BY B#
                    CALL D
                    END
```

```
 SUBROUTINE D          73/74    OPT=1                    FTN 4.8+528        81/10/01. 11.29.13        PAGE      1

1                SUBROUTINE D
                 PRINT *,# THIS MESSAGE COMES FROM SUBROUTINE D CALLED BY C#
                 CALL E
                 END
```

```
 SUBROUTINE E          73/74    OPT=1                    FTN 4.8+528        81/10/01. 11.29.13        PAGE      1

1                SUBROUTINE E
                 PRINT *,# THIS MESSAGE COMES FROM SUBROUTINE E CALLED BY D#
                 CALL F
                 END
```



```
 SUBROUTINE F          73/74    OPT=1                    FTN 4.8+528        81/10/01. 11.29.13        PAGE      1

1                SUBROUTINE F
                 PRINT *,# THIS MESSAGE COMES FROM SUBROUTINE F CALLED BY E#
           C     THIS IS COMMENT CARD NUMBER 1
           C     THIS IS COMMENT CARD NUMBER 2
5                I=2
                 J=4
                 K=I+J
           C     THIS IS COMMENT CARD NUMBER 3
           C     THIS IS COMMENT CARD NUMBER 4
10               PRINT *,I,J,K
                 CALL ERROR
           C     THIS IS COMMENT CARD NUMBER 5
                 END
```

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 6613

TRANSFER ADDRESS -- TEST 2173

PROGRAM ENTRY POINTS -- TEST 2173

********* ERROR SUMMARY

NE4100/// UNSATISFIED EXTERNAL REF -- ERROR

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| TEST | 111 | 2077 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | PROGRAM OPT=1 |
| A | 2210 | 20 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | SUBROUTINEOPT=1 |
| B | 2230 | 17 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | SUBROUTINEOPT=1 |
| C | 2247 | 17 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | SUBROUTINEOPT=1 |
| D | 2266 | 17 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | SUBROUTINEOPT=1 |
| E | 2305 | 17 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | SUBROUTINEOPT=1 |
| F | 2324 | 34 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | SUBROUTINEOPT=1 |
| SYSAID= | 2360 | 1 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2361 | 1 | | | | | | | |
| /FCL.C./ | 2362 | 27 | | | | | | | |
| /Q8.IO./ | 2411 | 127 | | | | | | | |
| Q2NTRY= | 2540 | 0 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2540 | 41 | | | | | | | |
| COMIO= | 2601 | 33 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2634 | 40 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL CAPSULE LOADING |
| FECMSK= | 2674 | 41 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2735 | 3 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2740 | 311 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON FLOATING OUTPUT CODE |
| FORSYS= | 3251 | 376 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 3647 | 25 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL MISC. UTILITIES. |
| GETFIT= | 3674 | 74 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| LDOUT= | 3770 | 240 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 4230 | 154 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON OUTPUT CODE |
| OUTF= | 4404 | 172 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 4576 | 5 | SL-SYSLIB | 81/09/18 | COMPASS | 3.6 | 552 | | 79/05/10. 81/08/07. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 4603 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 4643 | 14 | | | | | | | |
| FDL.RES | 4657 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 5070 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 5312 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| ERR$RM | 5747 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 5774 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 6063 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 6070 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 6252 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 6260 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 6274 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|
| CMF.GSS | 6332 | 22 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 6354 | 7 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | |
| CMM.R | 6363 | 206 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 6571 | 22 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

.380 CP SECONDS  240008 CM STORAGE USED  6 TABLE MOVES

```
 THIS MESSAGE COMES FROM PROGRAM TEST
 THIS MESSAGE COMES FROM SUBROUTINE A CALLED BY TEST
 THIS MESSAGE COMES FROM SUBROUTINE B CALLED BY A
 THIS MESSAGE COMES FROM SUBROUTINE C CALLED BY B
 THIS MESSAGE COMES FROM SUBROUTINE D CALLED BY C
 THIS MESSAGE COMES FROM SUBROUTINE E CALLED BY D
 THIS MESSAGE COMES FROM SUBROUTINE F CALLED BY E
2 4 6
```

```
                    VERSION = NOS 4F-FNT100          81/10/01. 11.29.23.
P          0  A0         0  B0         0   (A0) 0001 4023 3600 0000 0000
RA    227400  A1      2431  B1         1   (A1) 0000 0000 0045 4000 0000
FL     10700  A2      2410  B2      1571   (A2) 0000 0000 0000 0000 0000
EM      7007  A3      2456  B3       233   (A3) 0000 0000 0000 0000 2410
RAE        0  A4      7660  B4        11   (A4) 5555 5555 5555 5555 5555
FLE        0  A5       232  B5        12   (A5) 5535 5537 5541 0000 0000
MA      1000  A6      2410  B6      4570   (A6) 0000 0000 0000 0000 0000
              A7      2443  B7         1   (A7) 2011 0000 0000 0000 2410

X0   0104 0002 3300 0006 3200
X1   0000 0000 0045 4000 0000
X2   0000 0000 0000 0000 0000
X3   0000 0000 0000 0000 2410
X4   0000 0000 0000 0000 0220
X5   0000 0000 0000 0000 0000
X6   5500 0000 0000 0000 0000
X7   2011 0000 0000 0000 2410

(RA)    0001 4023 3600 0000 0000
(RA+1)  0000 0000 0000 0000 0000


CM DUMP FROM    2276 TO    2376.      VERSION = NOS 4F-FNT100          81/10/01. 11.29.23.
  2274      00000 00000 00000 00111      00060 00000 61000 02277      00000 00000 00000 00000      55241 01123 55150 52323
  2300      01070 55503 17150 52355      06221 71555 23250 22217      25241 11605 55045 50301      14140 50455 02315 50355
  2304      00000 00000 00000 00000      05555 55555 55550 02306      04000 02273 00000 00000      51100 02313 46000 46000
  2310      01000 04420 00020 02305      01000 02325 00030 02305      04000 02306 46000 46000      00000 00000 00000 00111
  2314      00060 00000 61000 02316      00000 00000 00000 00000      55241 01123 55150 52323      01070 55503 17150 52355
  2320      06221 71555 23250 22217      25241 11605 55055 50301      14140 50455 02315 50455      00000 00000 00000 00000
  2324      06555 55555 55550 02325      04000 02312 00000 00000      51100 02337 46000 46000      01000 04420 00020 02324
  2330      71700 00002 71600 00004      51700 02347 71700 00006      51600 02350 51700 02351      51100 02342 46000 46000
  2334      01000 04420 00120 02324      01004 02335 00130 02324      04000 02325 46000 46000      00000 00000 00000 00111
  2340      00060 00000 61000 02352      00000 00000 00000 00000      00000 00000 00000 00111      00020 00000 01000 02347
  2344      00020 00000 01000 02350      00020 00000 01000 02351      00000 00000 00000 00000      00000 00000 00000 00002
  2350      00000 00000 00000 00004      00000 00000 00000 00006      55241 01123 55150 52323      01070 55503 17150 52355
  2354      06221 71555 23250 22217      25241 11605 55065 50301      14140 50455 02315 50555      00000 00000 00000 00000
  2360      01000 03477 00000 00000      04000 03406 61000 46000      55555 55555 55555 55555      40404 04040 40404 04040
  2364      11162 02524 00000 00000      17252 42025 24000 00000      17171 27432 14774 13155      20001 20726 42717 30565
  2370      00000 00000 00000 00000      00000 00000 00000 02170      00000 00000 00000 11610      00000 00000 00000 00000
  2374      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
```

```
CM DUMP FROM    2000 TO  · 2400.     VERSION = NOS 4F-FNT100        81/10/01. 11.29.24.
  2000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
        DUPLICATED LINES.
  2164      00000 00000 00000 00000      20010 00000 00000 02170      00000 00000 00000 11610      00000 00000 00000 00000
  2170      17252 42025 24000 00111      00000 00000 00000 00000      24052 32455 55550 02173      04000 02174 00000 00001
  2174      51100 02200 46000 46000      01000 04420 00020 02172      01000 02211 00030 02172      51100 02172 04000 03332
  2200      00000 00000 00000 00111      00060 00000 45000 02203      00000 00000 00000 00000      55241 01123 55150 52323
  2204      01070 55503 17150 52355      06221 71555 20221 70722      01155 52405 23245 55555      00000 00000 00000 00000
  2210      01555 55555 55550 02211      04000 02177 00000 00000      51100 02216 46000 46000      01000 04420 00020 02210
  2214      01000 02231 00030 02210      04000 02211 46000 46000      00000 00000 00000 00111      00060 00000 64000 02221
  2220      00000 00000 00000 00000      55241 01123 55150 52323      01070 55503 17150 52355      06221 71555 23250 22217
  2224      25241 11605 55015 50301      14140 50455 02315 52405      23245 55555 55555 55555      00000 00000 00000 00000
  2230      02555 55555 55550 02231      04000 02215 00000 00000      51100 02236 46000 46000      01000 04420 00020 02230
  2234      01000 02250 00030 02230      04000 02231 46000 46000      00000 00000 00000 00111      00060 00000 61000 02241
  2240      00000 00000 00000 00000      55241 01123 55150 52323      01070 55503 17150 52355      06221 71555 23250 22217
  2244      25241 11605 55025 50301      14140 50455 02315 50155      00000 00000 00000 00000      03555 55555 55550 02250
  2250      04000 02235 00000 00000      51100 02255 46000 46000      01000 04420 00020 02247      01000 02267 00030 02247
  2254      04000 02250 46000 46000      00000 00000 00000 00111      00060 00000 61000 02260      00000 00000 00000 00000
  2260      55241 01123 55150 52323      01070 55503 17150 52355      06221 71555 23250 22217      25241 11605 55035 50301
  2264      14140 50455 02315 50255      00000 00000 00000 00000      04555 55555 55550 02267      04000 02254 00000 00000
  2270      51100 02274 46000 46000      01000 04420 00020 02266      01000 02306 00030 02266      04000 02267 46000 46000
  2274      00000 00000 00000 00111      00060 00000 61000 02277      00000 00000 00000 00000      55241 01123 55150 52323
  2300      01070 55503 17150 52355      06221 71555 23250 22217      25241 11605 55045 50301      14140 50455 02315 50355
  2304      00000 00000 00000 00000      05555 55555 55550 02306      04000 02273 00000 00000      51100 02313 46000 46000
  2310      01000 04420 00020 02305      01000 02325 00030 02305      04000 02306 46000 46000      00000 00000 00000 00111
  2314      00060 00000 61000 02316      00000 00000 00000 00000      55241 01123 55150 52323      01070 55503 17150 52355
  2320      06221 71555 23250 22217      25241 11605 55055 50301      14140 50455 02315 50455      00000 00000 00000 00000
  2324      06555 55555 55550 02325      04000 02312 00000 00000      51100 02337 46000 46000      01000 04420 00020 02324
  2330      71700 00002 71600 00004      51700 02347 71700 00006      51600 02350 51700 02351      51100 02342 46000 46000
  2334      01000 04420 00120 02324      01004 02335 00130 02324      04000 02325 46000 46000      00000 00000 00000 00111
  2340      00060 00000 61000 02352      00000 00000 00000 00000      00000 00000 00000 00111      00020 00000 01000 02347
  2344      00020 00000 01000 02350      00020 00000 01000 02351      00000 00000 00000 00000      00000 00000 00000 00002
  2350      00000 00000 00000 00004      00000 00000 00000 00006      55241 01123 55150 52323      01070 55503 17150 52355
  2354      06221 71555 23250 22217      25241 11605 55065 50301      14140 50455 02315 50555      00000 00000 00000 00000
  2360      01000 03477 00000 00000      04000 03406 61000 46000      55555 55555 55555 55555      40404 04040 40404 04040
  2364      11162 02524 00000 00000      17252 42025 24000 00000      17171 27432 14774 13155      20001 20726 42717 30565
  2370      00000 00000 00000 00000      00000 00000 00000 02170      00000 00000 00000 11610      00000 00000 00000 00000
  2374      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
  2400      00000 00000 00000 00024      00000 00000 00000 00000      00000 00000 00000 07640      20000 00000 00000 01132
                                                  FULL  DAYFILE.  81/10/01. 11.29.24.*11.29.12* PAGE    1

11.29.12.FTNERR.
11.29.12.UCCR, 64020,      0.043KCDS.
11.29.13.USER,DRF2755,.
11.29.13.MAP(PART)
11.29.13.FTN,R=0,GO.
11.29.21.      .211 CP SECONDS COMPILATION TIME
11.29.21.LGO
11.29.22.     NON-FATAL LOADER ERRORS - SEE MAP
11.29.23. CPU ERROR EXIT AT  402336.
11.29.23. CM OUT OF RANGE.
11.29.23.EXIT.
11.29.23.DMP(2000,2400)
11.29.24.UEAD,       0.001KUNS.
11.29.24.UEMS,       4.641KUNS.
11.29.24.UECP,       0.657SECS.
11.29.24.AESR,       2.721UNTS.
11.29.24.$OUT(*/OP=E)
11.29.24. NO FILES PROCESSED.
11.29.24.$DAYFILE(OUTPUT,JT=D)
11.34.16.UCLP, 64,023,      0.832KLNS.
```

# EXERCISE 4

STUDY QUESTIONS

1. (true/false) The job card is in the first card of a batch job's card deck and can start in any column.

2. (true/false) The job card and other control cards may be continued, logically, from column 80 of one card to column one of the next.

3. (true/false) Both an USER and CHARGE card are required as the second and third cards of a batch job deck.

4. (true/false) The executive (CPMTR/PPMTR) processes control cards.

5. (true/false) There can only be $77_8$ control cards in the control card record.

6. Considering the way in which the scheduler rolls jobs, would the RTIME or CTIME control statements serve as counters for job step timing? Explain.

# EXERCISE 5

PROBLEM

1. Set up a job that will continue even though error occur for the first seven control statements, then at the eight control statement alter the job such that normal error checking occurs (i.e. search for an EXIT statement).

2. Do a LIMITS report of your user number.

3. Set up a job that will illustrate use of the switch command (optional).

# LESSON GUIDE 3

## FILE MANIPULATION UTILITIES

LESSON PREVIEW:

This lesson acquaints the student with the NOS file manipulation utilities and demonstrates their use.

REFERENCES:

NOS Version 1 Reference Manual, Volume 1, Section 7.

TRAINING AIDS:

Visuals V3-1 through V3-21.

PROJECTS:

Exercises 6 and 7 (problems).

OBJECTIVES:

Upon completion of this lesson, the student will be able to:

1. State the current position on a file before and after specific events.
2. Use the TDUMP and CATALOG control statements to obtain reports on file content and structure.
3. Use the file manipulation statements, REWIND, SKIPF, SKIPFB, SKIPR, BKSP, and SKIPEI to get desired position on a file.
4. Use the file copy utilities, COPY and COPY(BR, BF, CR, CF, EI, and SBF) to copy portions or all of one file to another.

OBJECTIVES (Continued):

5. Use the special purpose copy routines to extract portions of a file by symbolic attributes on GTR and COPYX statements.

6. Use the VERIFY control statement and the verify parameter on the COPY statement to certify the results of copying.

7. Use the LOCK statement to prevent writing on a file.

8. Use the file termination utilities, RENAME, DISPOSE, EVICT, RETURN, UNLOAD and OUT in a proper manner.

9. Use the RESEQ, PACK and SORT statements to manipulate line numbered text as per timesharing.

10. Use the proper carriage control character to format line printer output.

The central idea is about file structure positions and content.

How files are created is discussed later in the lessons on program libraries and record management.

File content is unimportant when using some of the utilities but important with others. These differences will be examined in this section.

A file, as defined in lesson 1, is a collection of information and is manipulable by the file utilities. Files are more efficiently processed on disk because of the higher transfer rate of disk over tape† and, where possible, should be considered to be archival and not be physically positioned, but rather, a disk copy be used for sorts and searches.

Beginning of information (BOI), end-of-record (EOR), end-of-file (EOF) and end-of-information (EOI) are the lowest data organizational level at which the utilities work.

When is a file at BOI? Always, in the case of a null file (only an entry in the FNT) and only system sector and an EOI sector on the file chain . . . after a REWIND . . . and in many cases, as the enabling of a rewind parameter of a utility program, such as, "CATALOG,EFFILE, R."

- Make a list of all the system utility programs which use some sort of explicit rewind function.

When is a file at EOI? The null file is one example. Any other case is a situation in which a file has specifically been positioned to EOI by means of COPYEI, SKIPEI or as the result of some move on the files current position pointer.

Any other position on a file requires explicit programmer knowledge of the file structure. It is crucial therefore, to understand the effect of each control statement on the file position.

The utility programs are discussed and illustrated in nine functionally related groups.

The first group, File Structure and Content, gives a report about what is on a file and how it is organized. This is useful when a detailed analysis of the file is required and no other documentation about the file exists.

The second group, File Positioning Utilities can be used once the file structure and content are known, to position the file exactly to where a read or a write are to be performed on it.

---

† Another contributing factor for disk's greater efficiency is that disk files are managed by logically ordered chains.

The third group, General Copy Utilities, duplicates logical records and files or entire physical files, depending on the particular utility used and the parameters selected. File position is affected by the copies.

The fourth group, Special Copy Utilities, allows a lower level of control which allows search of a file for a record on logical attributes of type and name.

The fifth group, File Certification Utilities, provides a means to guarantee that information was duplicated accurately.

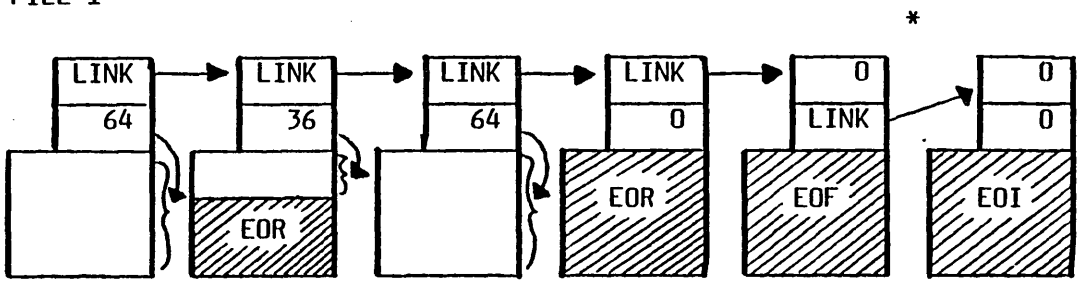
The sixth group, File Protection Utilities, provides a file sharing function on a controlled basis by preventing writing on a file.

The seventh group, File Termination Utilities, allows clearing of file space and names, and routing files to remote locations.

The eighth group, Line Number Control Utilities, are used primarily by time sharing users who are editing text which needs to be conditioned for compilation and execution.

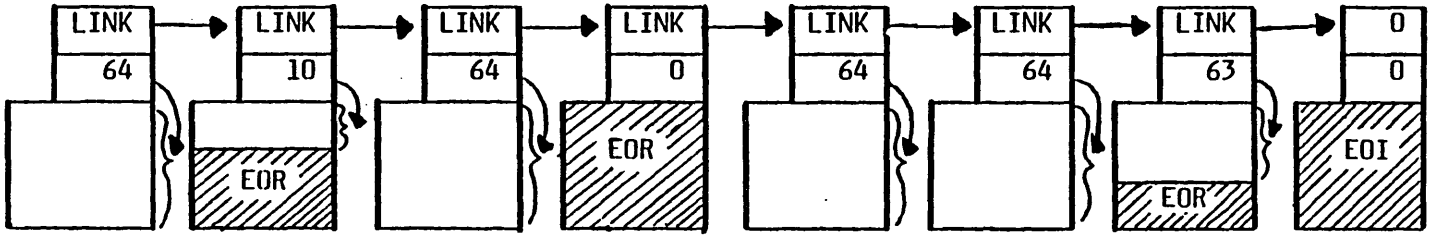
The last group, Dump/Load Central Memory Utilities, provides an alternate means of setting up a control point for program execution.

FILE 1

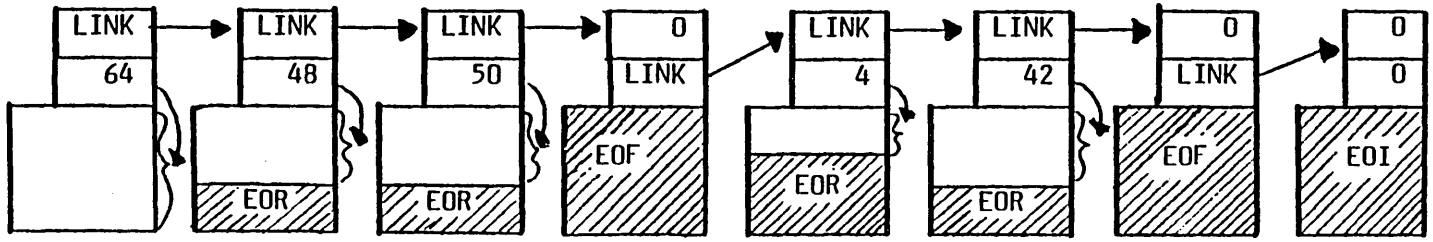



*

FILE 2




FILE 3




*Note that an EOF is not required as the last PRU of a file:

Examples of Mass Storage File Structure

```
MAKIT.  CREATE MULTIPLE FILE           1
USER,DRF2755,DRF.
COPYCR(INPUT,TESTCOP,2)
WRITEF(TESTCOP)
COPYCR(INPUT,TESTCOP,2)
WRITEF(TESTCOP)
COPYCR,,TESTCOP.
WRITEF(TESTCOP)
COPYCR,,TESTCOP,3)
WRITEF(TESTCOP)
SAVE(TESTCOP)  MAKE FILE PERMANENT
R  7/8/9 END-OF-RECORD
0AREC      FILE 1     RECORD 1   LINE 1
 BREC      FILE 1     RECORD 1    LINE 2
 CREC      FILE 1     RECORD 1     LINE 3
 DREC      FILE 1     RECORD 1      LINE 4
R  7/8/9 END-OF-RECORD
0EREC      FILE 1     RECORD 2  LINE 1
 FREC      FILE 1     RECORD 2   LINE 2
 GREC      FILE 1     RECORD 2    LINE 3
R  7/8/9 END-OF-RECORD
0HREC      FILE 2     RECORD 1  LINE 1
 IREC      FILE 2     RECORD 1   LINE 2
 JREC      FILE 2     RECORD 1    LINE 3
R  7/8/9 END-OF-RECORD
0KREC      FILE 2     RECORD 2  LINE 1
 LREC      FILE 2     RECORD 2   LINE 2
 MREC      FILE 2     RECORD 2    LINE 3
R  7/8/9 END-OF-RECORD
0NREC      FILE 3     RECORD 1  LINE 1
 OREC      FILE 3     RECORD 1   LINE 2
 PREC      FILE 3     RECORD 1    LINE 3
R  7/8/9 END-OF-RECORD
0QREC      FILE 4     RECORD 1  LINE 1
 RREC      FILE 4     RECORD 1   LINE 2
 SREC      FILE 4     RECORD 1    LINE 3
 TREC      FILE 4     RECORD 1     LINE 4
 UREC      FILE 4     RECORD 1      LINE 5
R  7/8/9 END-OF-RECORD
0VREC      FILE 4     RECORD 2  LINE 1
 WREC      FILE 4     RECORD 2   LINE 2
R  7/8/9 END-OF-RECORD
0XREC      FILE 4     RECORD 3  LINE 1
 YREC      FILE 4     RECORD 3   LINE 2
 ZREC      FILE 4     RECORD 3    LINE 3
Q  6/7/8/9 END-OF-INFORMATION
```



```
                FULL  DAYFILE.  81/10/01. 11.31.52.*11.31.50* PAGE
     11.31.50.MAKIT.  CREATE MULTIPLE FILE           1
     11.31.50.UCCR, 64020,      0.037KCDS.
     11.31.50.USER,DRF2755,.
     11.31.50.COPYCR(INPUT,TESTCOP,2)
     11.31.50. COPY COMPLETE.
     11.31.50.WRITEF(TESTCOP)
     11.31.50.COPYCR(INPUT,TESTCOP,2)
     11.31.50. COPY COMPLETE.
AREC 11.31.50.WRITEF(TESTCOP)
BREC 11.31.51.COPYCR,,TESTCOP.
CREC 11.31.51. COPY COMPLETE.
DREC 11.31.51.WRITEF(TESTCOP)
     11.31.51.COPYCR,,TESTCOP,3)
EREC 11.31.51. COPY COMPLETE.
FREC 11.31.51.WRITEF(TESTCOP)
GREC 11.31.51.SAVE(TESTCOP)  MAKE FILE PERMANENT
     11.31.52.UEAD,       0.001KUNS.
HREC 11.31.52.UEPF,       0.009KUNS.
IREC 11.31.52.UEMS,       0.170KUNS.
JREC 11.31.52.UECP,       0.038SECS.
     11.31.52.AESR,       1.058UNTS.
KREC 11.31.52.$OUT(*/OP=E)
LREC 11.31.52. NO FILES PROCESSED.
MREC 11.31.52.$DAYFILE(OUTPUT,JT=D)
     11.34.50.UCLP, 64,023,      0.128KLNS.
NREC
OREC
PREC
QREC
RREC
SREC
TREC
UREC
VREC
WREC
XREC
YREC
ZREC
```

```
SEEIT.   EXAMINE FILE CREATED          2
USER,DRF2755,DRF.
GET(TESTCOP)
CATALOG(TESTCOP,N,R)
TDUMP(I=TESTCOP)
Q  6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/01. 11.34.54.*11.34.45* PAGE    1

11.34.45.SEEIT.   EXAMINE FILE CREATED          2
11.34.45.UCCR, 64020,      0.005KCDS.
11.34.46.USER,DRF2755,.
11.34.46.GET(TESTCOP)
11.34.47.CATALOG(TESTCOP,N,R)
11.34.49. CATALOG COMPLETE.
11.34.49.TDUMP(I=TESTCOP)
11.34.52. TDUMP COMPLETE.
11.34.53.UEAD,       0.001KUNS.
11.34.53.UEPF,       0.009KUNS.
11.34.53.UEMS,       0.268KUNS.
11.34.53.UECP,       0.247SECS.
11.34.53.AESR,       1.287UNTS.
11.34.53.$OUT(*/OP=E)
11.34.54. NO FILES PROCESSED.
11.34.54.$DAYFILE(OUTPUT,JT=D)
11.35.57.UCLP, 64,023,      0.704KLNS.
```

1

2

CATALOG OF TESTCOP FILE 1 81/10/01. 11.34.47. PAGE 1

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | OAREC | TEXT | 44 | 6026 | | |
| 2 | OEREC | TEXT | 33 | 2610 | | |
| 3 | * EOF * | SUM = | 77 | | | |

CATALOG OF TESTCOP FILE 2 81/10/01. 11.34.47. PAGE 2

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | OHREC | TEXT | 33 | 6041 | | |
| 2 | OKREC | TEXT | 33 | 3313 | | |
| 3 | * EOF * | SUM = | 66 | | | |

CATALOG OF TESTCOP FILE 3 81/10/01. 11.34.47. PAGE 3

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | ONREC | TEXT | 33 | 0065 | | |
| 2 | * EOF * | SUM = | 33 | | | |

CATALOG OF TESTCOP FILE 4 81/10/01. 11.34.47. PAGE 4

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | OQREC | TEXT | 55 | 2302 | | |
| 2 | OVREC | TEXT | 22 | 5374 | | |
| 3 | OXREC | TEXT | 33 | 3604 | | |
| 4 | * EOF * | SUM = | 132 | | | |

CATALOG OF TESTCOP FILE 5 81/10/01. 11.34.47. PAGE 5

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| | * EOI * | SUM = | 0 | | | |

```
F  1 R   1 W     0- 3301 2205 0355 5555 5555   0611 1405 5534 5555 5555   2205 0317 2204 5534 5555   5514 1116 0555 3455 5555
                    0 A  R E  C                   F I  L E    1              R E  C O  R D    1             L  I N  E    1
F  1 R   1 W     4- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 0122 0503
                                                                                                                  A R  E C
F  1 R   1 W    10- 0000 0000 0000 0000 0000   5502 2205 0355 5555 5555   0611 1405 5534 5555 5555   2205 0317 2204 5534 5555
                                                   B  R E  C                   F I  L E    1              R E  C O  R D    1
F  1 R   1 W    14- 5555 1411 1605 5535 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      L I  N E    2
F  1 R   1 W    20- 5555 5555 5555 0222 0503   0000 0000 0000 0000 0000   5503 2205 0355 5555 5555   0611 1405 5534 5555 5555
                                   B R  E C                                    C  R E  C                   F I  L E    1
F  1 R   1 W    24- 2205 0317 2204 5534 5555   5555 5514 1116 0555 3655   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                    R E  C O  R D    1                  L  I N  E    3
F  1 R   1 W    30- 5555 5555 5555 5555 5555   5555 5555 5555 0322 0503   0000 0000 0000 0000 0000   5504 2205 0355 5555 5555
                                                                C R  E C                                    D  R E  C
F  1 R   1 W    34- 0611 1405 5534 5555 5555   2205 0317 2204 5534 5555   5555 5555 1411 1605 5537   5555 5555 5555 5555 5555
                    F I  L E    1              R E  C O  R D    1                        L  I N  E    4
F  1 R   1 W    40- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 0422 0503   0000 0000 0000 0000 0000
                                                                                        D R  E C

-- END OF RECORD --

F  1 R   2 W     0- 3305 2205 0355 5555 5555   0611 1405 5534 5555 5555   2205 0317 2204 5535 5555   1411 1605 5534 5555 5555
                    0 E  R E  C                   F I  L E    1              R E  C O  R D    2             L  I N  E    1
F  1 R   2 W     4- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 0522 0503
                                                                                                                  E R  E C
F  1 R   2 W    10- 0000 0000 0000 0000 0000   5506 2205 0355 5555 5555   0611 1405 5534 5555 5555   2205 0317 2204 5535 5555
                                                   F  R E  C                   F I  L E    1              R E  C O  R D    2
F  1 R   2 W    14- 5514 1116 0555 3555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      L  I N  E    2
F  1 R   2 W    20- 5555 5555 5555 0622 0503   0000 0000 0000 0000 0000   5507 2205 0355 5555 5555   0611 1405 5534 5555 5555
                                   F R  E C                                    G  R E  C                   F I  L E    1
F  1 R   2 W    24- 2205 0317 2204 5535 5555   5555 1411 1605 5536 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                    R E  C O  R D    2                  L  I N  E    3
F  1 R   2 W    30- 5555 5555 5555 5555 5555   5555 5555 5555 0722 0503   0000 0000 0000 0000 0000
                                                                G R  E C

-- END OF RECORD --

-- END OF FILE --

F  2 R   1 W     0- 3310 2205 0355 5555 5555   0611 1405 5535 5555 5555   2205 0317 2204 5534 5555   1411 1605 5534 5555 5555
                    0 H  R E  C                   F I  L E    2              R E  C O  R D    1             L  I N  E    1
F  2 R   1 W     4- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 1022 0503
                                                                                                                  H R  E C
F  2 R   1 W    10- 0000 0000 0000 0000 0000   5511 2205 0355 5555 5555   0611 1405 5535 5555 5555   2205 0317 2204 5534 5555
                                                   I  R E  C                   F I  L E    2              R E  C O  R D    1
F  2 R   1 W    14- 5514 1116 0555 3555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      L  I N  E    2
F  2 R   1 W    20- 5555 5555 5555 1122 0503   0000 0000 0000 0000 0000   5512 2205 0355 5555 5555   0611 1405 5535 5555 5555
                                   I R  E C                                    J  R E  C                   F I  L E    2
F  2 R   1 W    24- 2205 0317 2204 5534 5555   5555 1411 1605 5536 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                    R E  C O  R D    1                  L  I N  E    3
F  2 R   1 W    30- 5555 5555 5555 5555 5555   5555 5555 5555 1222 0503   0000 0000 0000 0000 0000
                                                                J R  E C

-- END OF RECORD --

F  2 R   2 W     0- 3313 2205 0355 5555 5555   0611 1405 5535 5555 5555   2205 0317 2204 5535 5555   1411 1605 5534 5555 5555
                    0 K  R E  C                   F I  L E    2              R E  C O  R D    2             L  I N  E    1
F  2 R   2 W     4- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 1322 0503
                                                                                                                  K R  E C
```

```
F  2 R   2 W    10- 0000 0000 0000 0000 0000   5514 2205 0355 5555 5555   0611 1405 5535 5555 5555   2205 0317 2204 5535 5555
                                                    L  R  E  C                  F  I  L  E    2              R  E  C  O  R  D    2
F  2 R   2 W    14- 5514 1116 0555 3555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      L  I  N  E    2
F  2 R   2 W    20- 5555 5555 5555 1422 0503   0000 0000 0000 0000 0000   5515 2205 0355 5555 5555   0611 1405 5535 5555 5555
                                  L  R  E  C                                  M  R  E  C                  F  I  L  E    2
F  2 R   2 W    24- 2205 0317 2204 5535 5555   5555 1411 1605 5536 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      R  E  C  O  R  D    2                  L  I  N  E    3
F  2 R   2 W    30- 5555 5555 5555 5555 5555   5555 5555 5555 1522 0503   0000 0000 0000 0000 0000
                                                                M  R  E  C
-- END OF RECORD --

-- END OF FILE --

F  3 R   1 W     0- 3316 2205 0355 5555 5555   0611 1405 5536 5555 5555   2205 0317 2204 5534 5555   1411 1605 5534 5555 5555
                      O  N  R  E  C               F  I  L  E    3              R  E  C  O  R  D    1              L  I  N  E    1
F  3 R   1 W     4- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 1622 0503
                                                                                                                  N  R  E  C
F  3 R   1 W    10- 0000 0000 0000 0000 0000   5517 2205 0355 5555 5555   0611 1405 5536 5555 5555   2205 0317 2204 5534 5555
                                                    O  R  E  C                  F  I  L  E    3              R  E  C  O  R  D    1
F  3 R   1 W    14- 5514 1116 0555 3555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      L  I  N  E    2
F  3 R   1 W    20- 5555 5555 5555 1722 0503   0000 0000 0000 0000 0000   5520 2205 0355 5555 5555   0611 1405 5536 5555 5555
                                  O  R  E  C                                  P  R  E  C                  F  I  L  E    3
F  3 R   1 W    24- 2205 0317 2204 5534 5555   5555 1411 1605 5536 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      R  E  C  O  R  D    1                  L  I  N  E    3
F  3 R   1 W    30- 5555 5555 5555 5555 5555   5555 5555 5555 2022 0503   0000 0000 0000 0000 0000
                                                                P  R  E  C
-- END OF RECORD --

-- END OF FILE --

F  4 R   1 W     0- 3321 2205 0355 5555 5555   0611 1405 5537 5555 5555   2205 0317 2204 5534 5555   1411 1605 5534 5555 5555
                      O  Q  R  E  C               F  I  L  E    4              R  E  C  O  R  D    1              L  I  N  E    1
F  4 R   1 W     4- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 2122 0503
                                                                                                                  Q  R  E  C
F  4 R   1 W    10- 0000 0000 0000 0000 0000   5522 2205 0355 5555 5555   0611 1405 5537 5555 5555   2205 0317 2204 5534 5555
                                                    R  R  E  C                  F  I  L  E    4              R  E  C  O  R  D    1
F  4 R   1 W    14- 5514 1116 0555 3555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      L  I  N  E    2
F  4 R   1 W    20- 5555 5555 5555 2222 0503   0000 0000 0000 0000 0000   5523 2205 0355 5555 5555   0611 1405 5537 5555 5555
                                  R  R  E  C                                  S  R  E  C                  F  I  L  E    4
F  4 R   1 W    24- 2205 0317 2204 5534 5555   5555 1411 1605 5536 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      R  E  C  O  R  D    1                  L  I  N  E    3
F  4 R   1 W    30- 5555 5555 5555 5555 5555   5555 5555 5555 2322 0503   0000 0000 0000 0000 0000   5524 2205 0355 5555 5555
                                                                S  R  E  C                                  T  R  E  C
F  4 R   1 W    34- 0611 1405 5537 5555 5555   2205 0317 2204 5534 5555   5555 5514 1116 0555 3755   5555 5555 5555 5555 5555
                      F  I  L  E    4             R  E  C  O  R  D    1                  L  I  N  E    4
F  4 R   1 W    40- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 2422 0503   0000 0000 0000 0000 0000
                                                                                        T  R  E  C
F  4 R   1 W    44- 5525 2205 0355 5555 5555   0611 1405 5537 5555 5555   2205 0317 2204 5534 5555   5555 5555 1411 1605 5540
                      U  R  E  C                  F  I  L  E    4             R  E  C  O  R  D    1                  L  I  N  E    5
F  4 R   1 W    50- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 2522 0503
                                                                                                                  U  R  E  C
F  4 R   1 W    54- 0000 0000 0000 0000 0000

-- END OF RECORD --
```

```
F  4 R   2 W     0- 3326 2205 0355 5555 5555   0611 1405 5537 5555 5555   2205 0317 2204 5535 5555   1411 1605 5534 5555 5555
                     0 V  R E  C                F I  L E    4              R E  C O  R D    2         L I  N E    1
F  4 R   2 W     4- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 2622 0503
                                                                                                                V R  E C
F  4 R   2 W    10- 0000 0000 0000 0000 0000   5527 2205 0355 5555 5555   0611 1405 5537 5555 5555   2205 0317 2204 5535 5555
                                                  W  R E  C                  F I  L E    4              R E  C O  R D    2
F  4 R   2 W    14- 5514 1116 0555 3555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      L  I N  E    2
F  4 R   2 W    20- 5555 5555 5555 2722 0503   0000 0000 0000 0000 0000
                                   W R  E C
-- END OF RECORD --

F  4 R   3 W     0- 3330 2205 0355 5555 5555   0611 1405 5537 5555 5555   2205 0317 2204 5536 5555   1411 1605 5534 5555 5555
                     0 X  R E  C                F I  L E    4              R E  C O  R D    3         L I  N E    1
F  4 R   3 W     4- 5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 3022 0503
                                                                                                                X R  E C
F  4 R   3 W    10- 0000 0000 0000 0000 0000   5531 2205 0355 5555 5555   0611 1405 5537 5555 5555   2205 0317 2204 5536 5555
                                                  Y  R E  C                  F I  L E    4              R E  C O  R D    3
F  4 R   3 W    14- 5514 1116 0555 3555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                      L  I N  E    2
F  4 R   3 W    20- 5555 5555 5555 3122 0503   0000 0000 0000 0000 0000   5532 2205 0355 5555 5555   0611 1405 5537 5555 5555
                                   Y R  E C                                  Z  R E  C                  F I  L E    4
F  4 R   3 W    24- 2205 0317 2204 5536 5555   5555 1411 1605 5536 5555   5555 5555 5555 5555 5555   5555 5555 5555 5555 5555
                     R E  C O  R D    3                  L I  N E    3
F  4 R   3 W    30- 5555 5555 5555 5555 5555   5555 5555 5555 3222 0503   0000 0000 0000 0000 0000
                                                                 Z R  E C
-- END OF RECORD --

-- END OF FILE --

-- END OF INFORMATION --

-- END OF DUMP --
```

```
COPYIT. ILLUSTRATION OF COPY COMMANDS  3
USER,DRF2755,DRF.
GET(TESTCOP)
COPYCF(TESTCOP,OUTPUT,4)               1
REWIND(TESTCOP)
COPYBF(TESTCOP,OUTPUT.                 2
REWIND(TESTCOP)
COPYCF(TESTCOP,OUTPUT,1)               3
REWIND(TESTCOP)
COPYCR(TESTCOP)                        4
COPYBR(TESTCOP)                        5
REWIND(TESTCOP)
COPYCR(TESTCOP,OUTPUT,10)              6
REWIND(TESTCOP)
COPYCR(TESTCOP,OUTPUT,2,37,79)         7
REWIND(TESTCOP)
COPYCR(TESTCOP,TESTCOP)
COPYCR(TESTCOP,OUTPUT)                 8
REWIND(TESTCOP)
COPYCF(TESTCOP,TESTCOP,1)
COPYSBF(TESTCOP,OUTPUT,2)              9
REWIND(TESTCOP)
COPYCF(TESTCOP,OTHER)
WRITEF(OTHER)
COPYCF(TESTCOP,OTHER)
REWIND(OTHER)
COPY(OTHER,OUTPUT)                     10
CATALOG(OTHER,N,R)                     11
REWIND(OTHER)
COPYEI(OTHER,OUTPUT)                   12
Q  6/7/8/9 END-OF-INFORMATION
```

3-11

```
     FULL  DAYFILE.  81/10/01. 11.35.16.*11.34.54* PAGE    1

11.34.54.COPYIT. ILLUSTRATION OF COPY COMMANDS  3
11.34.54.UCCR, 64020,      0.030KCDS.
11.34.54.USER,DRF2755,.
11.34.54.GET(TESTCOP)
11.34.55.COPYCF(TESTCOP,OUTPUT,4)                1
11.34.58. COPY COMPLETE.
11.34.59.REWIND(TESTCOP)
11.34.59.COPYBF(TESTCOP,OUTPUT.                  2
11.35.00. COPY COMPLETE.
11.35.00.REWIND(TESTCOP)
11.35.00.COPYCF(TESTCOP,OUTPUT,1)                3
11.35.01. COPY COMPLETE.
11.35.01.REWIND(TESTCOP)
11.35.01.COPYCR(TESTCOP)                         4
11.35.02. COPY COMPLETE.
11.35.02.COPYBR(TESTCOP)                         5
11.35.02. COPY COMPLETE.
11.35.02.REWIND(TESTCOP)
11.35.03.COPYCR(TESTCOP,OUTPUT,10)               6
11.35.06. COPY COMPLETE.
11.35.06.REWIND(TESTCOP)
11.35.07.COPYCR(TESTCOP,OUTPUT,2,37,79)          7
11.35.08. COPY COMPLETE.
11.35.08.REWIND(TESTCOP)
11.35.08.COPYCR(TESTCOP,TESTCOP)
11.35.08. COPY COMPLETE.
11.35.08.COPYCR(TESTCOP,OUTPUT)                  8
11.35.08. COPY COMPLETE.
11.35.09.REWIND(TESTCOP)
11.35.09.COPYCF(TESTCOP,TESTCOP,1)
11.35.09. COPY COMPLETE.
11.35.10.COPYSBF(TESTCOP,OUTPUT,2)               9
11.35.11. COPY COMPLETE.
11.35.11.REWIND(TESTCOP)
11.35.11.COPYCF(TESTCOP,OTHER)
11.35.12. COPY COMPLETE.
11.35.12.WRITEF(OTHER)
11.35.12.COPYCF(TESTCOP,OTHER)
11.35.13. COPY COMPLETE.
11.35.13.REWIND(OTHER)
11.35.13.COPY(OTHER,OUTPUT)                      10
11.35.14. COPY COMPLETE.
11.35.14.CATALOG(OTHER,N,R)                      11
11.35.15. CATALOG COMPLETE.
11.35.15.REWIND(OTHER)
11.35.15.COPYEI(OTHER,OUTPUT)                    12
11.35.16. EOI ENCOUNTERED.
11.35.16.UEAD,       0.001KUNS.
11.35.16.UEPF,       0.009KUNS.
11.35.16.UEMS,       0.850KUNS.
11.35.16.UFCP,       0.192SECS.
11.35.16.AESR,       1.300UNTS.
11.35.16.$OUT(*/OP=E)
11.35.16. NO FILES PROCESSED.
11.35.16.$DAYFILE(OUTPUT,JT=D)
11.36.56.UCLP, 64,023,      0.704KLNS.
```

```
AREC    FILE 1    RECORD 1   LINE 1                    AREC
BREC    FILE 1    RECORD 1    LINE 2                   BREC
CREC    FILE 1    RECORD 1     LINE 3                  CREC
DREC    FILE 1    RECORD 1      LINE 4                 DREC

EREC    FILE 1    RECORD 2  LINE 1                     EREC
FREC    FILE 1    RECORD 2   LINE 2                    FREC
GREC    FILE 1    RECORD 2    LINE 3                   GREC

HREC    FILE 2    RECORD 1  LINE 1                     HREC
IREC    FILE 2    RECORD 1   LINE 2                    IREC
JREC    FILE 2    RECORD 1    LINE 3                   JREC

KREC    FILE 2    RECORD 2  LINE 1                     KREC
LREC    FILE 2    RECORD 2   LINE 2                    LREC
MREC    FILE 2    RECORD 2    LINE 3                   MREC

NREC    FILE 3    RECORD 1  LINE 1                     NREC
OREC    FILE 3    RECORD 1   LINE 2                    OREC   1
PREC    FILE 3    RECORD 1    LINE 3                   PREC

QREC    FILE 4    RECORD 1  LINE 1                     QREC
RREC    FILE 4    RECORD 1   LINE 2                    RREC
SREC    FILE 4    RECORD 1    LINE 3                   SREC
TREC    FILE 4    RECORD 1     LINE 4                  TREC
UREC    FILE 4    RECORD 1      LINE 5                 UREC

VREC    FILE 4    RECORD 2  LINE 1                     VREC
WREC    FILE 4    RECORD 2   LINE 2                    WREC

XREC    FILE 4    RECORD 3  LINE 1                     XREC
YREC    FILE 4    RECORD 3   LINE 2                    YREC
ZREC    FILE 4    RECORD 3    LINE 3                   ZREC
```

---

```
AREC    FILE 1    RECORD 1   LINE 1                    AREC
BREC    FILE 1    RECORD 1    LINE 2                   BREC
CREC    FILE 1    RECORD 1     LINE 3                  CREC
DREC    FILE 1    RECORD 1      LINE 4                 DREC   2

EREC    FILE 1    RECORD 2  LINE 1                     EREC
FREC    FILE 1    RECORD 2   LINE 2                    FREC
GREC    FILE 1    RECORD 2    LINE 3                   GREC
```

---

```
AREC    FILE 1    RECORD 1   LINE 1                    AREC
BREC    FILE 1    RECORD 1    LINE 2                   BREC
CREC    FILE 1    RECORD 1     LINE 3                  CREC
DREC    FILE 1    RECORD 1      LINE 4                 DREC   3

EREC    FILE 1    RECORD 2  LINE 1                     EREC
FREC    FILE 1    RECORD 2   LINE 2                    FREC
GREC    FILE 1    RECORD 2    LINE 3                   GREC
```

---

```
AREC    FILE 1    RECORD 1   LINE 1                    AREC
BREC    FILE 1    RECORD 1    LINE 2                   BREC
CREC    FILE 1    RECORD 1     LINE 3                  CREC   4
DREC    FILE 1    RECORD 1      LINE 4                 DREC
```

---

```
EREC    FILE 1    RECORD 2  LINE 1                     EREC
FREC    FILE 1    RECORD 2   LINE 2                    FREC   5
GREC    FILE 1    RECORD 2    LINE 3                   GREC
```

---

```
AREC    FILE 1    RECORD 1   LINE 1                    AREC
BREC    FILE 1    RECORD 1    LINE 2                   BREC
CREC    FILE 1    RECORD 1     LINE 3                  CREC
DREC    FILE 1    RECORD 1      LINE 4                 DREC
```

```
EREC     FILE 1    RECORD 2  LINE 1                                  EREC
FREC     FILE 1    RECORD 2   LINE 2                                 FREC
GREC     FILE 1    RECORD 2    LINE 3                                GREC

HREC     FILE 2    RECORD 1  LINE 1                                  HREC
IREC     FILE 2    RECORD 1   LINE 2                                 IREC
JREC     FILE 2    RECORD 1    LINE 3                                JREC

KREC     FILE 2    RECORD 2  LINE 1                                  KREC
LREC     FILE 2    RECORD 2   LINE 2                                 LREC
MREC     FILE 2    RECORD 2    LINE 3                                MREC

NREC     FILE 3    RECORD 1  LINE 1                                  NREC
OREC     FILE 3    RECORD 1   LINE 2                                 OREC
PREC     FILE 3    RECORD 1    LINE 3                                PREC

QREC     FILE 4    RECORD 1  LINE 1                                  QREC
RREC     FILE 4    RECORD 1   LINE 2                                 RREC
SREC     FILE 4    RECORD 1    LINE 3                                SREC
TREC     FILE 4    RECORD 1     LINE 4                               TREC
UREC     FILE 4    RECORD 1      LINE 5                              UREC

VREC     FILE 4    RECORD 2  LINE 1                                  VREC
WREC     FILE 4    RECORD 2   LINE 2                                 WREC
```

6

7

ARE
BRE
CRE
DRE
ERE

2
3
E 4

```
EREC     FILE 1    RECORD 2  LINE 1
FREC     FILE 1    RECORD 2   LINE 2
GREC     FILE 1    RECORD 2    LINE 3
```

```
EREC
FREC
GREC
```

8

```
OHREC     FILE 2    RECORD 1  LINE 1                                  HREC
 IREC     FILE 2    RECORD 1   LINE 2                                 IREC
 JREC     FILE 2    RECORD 1    LINE 3                                JREC


 OKREC    FILE 2    RECORD 2  LINE 1                                  KREC
  LREC    FILE 2    RECORD 2   LINE 2                                 LREC
  MREC    FILE 2    RECORD 2    LINE 3                                MREC
```

9

```
ONREC     FILE 3    RECORD 1  LINE 1                                  NREC
 OREC     FILE 3    RECORD 1   LINE 2                                 OREC
 PREC     FILE 3    RECORD 1    LINE 3                                PREC
```

```
AREC      FILE 1    RECORD 1   LINE 1                                  AREC
BREC      FILE 1    RECORD 1    LINE 2                                 BREC
CREC      FILE 1    RECORD 1     LINE 3                               CREC
DREC      FILE 1    RECORD 1      LINE 4                              DREC

EREC      FILE 1    RECORD 2  LINE 1                                  EREC
FREC      FILE 1    RECORD 2   LINE 2                                 FREC
GREC      FILE 1    RECORD 2    LINE 3                                GREC
```

10

CATALOG OF OTHER FILE 1 81/10/01. 11.35.14. PAGE 1

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | OAREC | TEXT | 44 | 6026 | | |
| 2 | OEREC | TEXT | 33 | 2610 | | |
| 3 | * EOF * | SUM = | 77 | | | |

CATALOG OF OTHER FILE 2 81/10/01. 11.35.14. PAGE 2

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | * EOF * | SUM = | 0 | | | |

CATALOG OF OTHER FILE 3 81/10/01. 11.35.14. PAGE 3

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | OHREC | TEXT | 33 | 6041 | | |
| 2 | OKREC | TEXT | 33 | 3313 | | |
| 3 | * EOF * | SUM = | 66 | | | |



CATALOG OF OTHER FILE 4 81/10/01. 11.35.14. PAGE 4

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| | * EOI * | SUM = | 0 | | | |

| | | | | |
|---|---|---|---|---|
| AREC | FILE 1 | RECORD 1 | LINE 1 | AREC |
| BREC | FILE 1 | RECORD 1 | LINE 2 | BREC |
| CREC | FILE 1 | RECORD 1 | LINE 3 | CREC |
| DREC | FILE 1 | RECORD 1 | LINE 4 | DREC |
| EREC | FILE 1 | RECORD 2 | LINE 1 | EREC |
| FREC | FILE 1 | RECORD 2 | LINE 2 | FREC |
| GREC | FILE 1 | RECORD 2 | LINE 3 | GREC |
| HREC | FILE 2 | RECORD 1 | LINE 1 | HREC |
| IREC | FILE 2 | RECORD 1 | LINE 2 | IREC |
| JREC | FILE 2 | RECORD 1 | LINE 3 | JREC |
| KREC | FILE 2 | RECORD 2 | LINE 1 | KREC |
| LREC | FILE 2 | RECORD 2 | LINE 2 | LREC |
| MREC | FILE 2 | RECORD 2 | LINE 3 | MREC |

```
SKIPIT.  FILE POSITIONING ON.TESTCOP    4
USER,DRF2755,DRF.
GET(TESTCOP)
COPYCR(TESTCOP,OUTPUT)                  1
REWIND(TESTCOP)
SKIPR(TESTCOP,1)
COPYCR(TESTCOP,OUTPUT)                  2
BKSP(TESTCOP,1)
COPYCR(TESTCOP,OUTPUT)                  3
REWIND(TESTCOP)
SKIPF(TESTCOP,3)
COPYCR(TESTCOP)                         4
SKIPFB(TESTCOP,1)
COPYCR(TESTCOP,OUTPUT)                  5
REWIND(TESTCOP)
COPYCR(TESTCOP)                         6
SKIPEI(TESTCOP)
BKSP(TESTCOP,1)
COPYCR(TESTCOP,OUTPUT)
BKSP(TESTCOP,3)
COPYCR(TESTCOP)                         7
SKIPFB(TESTCOP,2)
COPYCR(TESTCOP,OUTPUT.                  8
REWIND(TESTCOP)
SKIPR(TESTCOP,4)
COPYCR(TESTCOP)                         9
BKSP(TESTCOP,4)
COPYCR(TESTCOP)                        10
REWIND(TESTCOP)
SKIPR(TESTCOP,2)
COPYCR(TESTCOP)
Q  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/01. 11.35.10.*11.34.54* PAGE    1

11.34.54.SKIPIT.  FILE POSITIONING ON TESTCOP    4
11.34.54.UCCR, 64020,      0.031KCDS.
11.34.54.USER,DRF2755,.
11.34.55.GET(TESTCOP)
11.34.56.COPYCR(TESTCOP,OUTPUT)                  1
11.34.56. COPY COMPLETE.
11.34.56.REWIND(TESTCOP)
11.34.56.SKIPR(TESTCOP,1)
11.34.57.COPYCR(TESTCOP,OUTPUT)                  2
11.34.57. COPY COMPLETE.
11.34.57.BKSP(TESTCOP,1)
11.34.58.COPYCR(TESTCOP,OUTPUT)                  3
11.34.58. COPY COMPLETE.
11.34.59.REWIND(TESTCOP)
11.34.59.SKIPF(TESTCOP,3)
11.34.59.COPYCR(TESTCOP)                         4
11.35.00. COPY COMPLETE.
11.35.00.SKIPFB(TESTCOP,1)
11.35.01.COPYCR(TESTCOP,OUTPUT)                  5
11.35.01. COPY COMPLETE.
11.35.01.REWIND(TESTCOP)
11.35.01.COPYCR(TESTCOP)                         6
11.35.01. COPY COMPLETE.
11.35.02.SKIPEI(TESTCOP)
11.35.02.BKSP(TESTCOP,1)
11.35.02.COPYCR(TESTCOP,OUTPUT)
11.35.03. COPY COMPLETE.
11.35.03.BKSP(TESTCOP,3)
11.35.04.COPYCR(TESTCOP)                         7
11.35.04. COPY COMPLETE.
11.35.05.SKIPFB(TESTCOP,2)
11.35.05.COPYCR(TESTCOP,OUTPUT.                  8
11.35.05. COPY COMPLETE.
11.35.06.REWIND(TESTCOP)
11.35.06.SKIPR(TESTCOP,4)
11.35.06.COPYCR(TESTCOP)                         9
11.35.07. COPY COMPLETE.
11.35.07.BKSP(TESTCOP,4)
11.35.07.COPYCR(TESTCOP)                        10
11.35.08. COPY COMPLETE.
11.35.08.REWIND(TESTCOP)
11.35.08.SKIPR(TESTCOP,2)
11.35.08.COPYCR(TESTCOP)
11.35.08. COPY COMPLETE.
11.35.09.UEAD,       0.001KUNS.
11.35.09.UEPF,       0.009KUNS.
11.35.09.UEMS,       0.432KUNS.
11.35.09.UECP,       0.102SECS.
11.35.09.AESR,       1.154UNTS.
11.35.09.$OUT(*/OP=E)
11.35.09. NO FILES PROCESSED.
11.35.10.$DAYFILE(OUTPUT,JT=D)
11.36.45.UCLP, 64,023,       0.128KLNS.
```

```
AREC     FILE 1    RECORD 1   LINE 1                    AREC
BREC     FILE 1    RECORD 1    LINE 2                   BREC   1
CREC     FILE 1    RECORD 1     LINE 3                  CREC
DREC     FILE 1    RECORD 1      LINE 4                 DREC
```

```
EREC     FILE 1    RECORD 2  LINE 1                     EREC
FREC     FILE 1    RECORD 2   LINE 2                    FREC   2
GREC     FILE 1    RECORD 2    LINE 3                   GREC
```

```
EREC     FILE 1    RECORD 2  LINE 1                     EREC
FREC     FILE 1    RECORD 2   LINE 2                    FREC   3
GREC     FILE 1    RECOPD 2    LINE 3                   GREC
```

```
QREC     FILE 4    RECORD 1  LINE 1                     QREC
RREC     FILE 4    RECORD 1   LINE 2                    RREC
SREC     FILE 4    RECORD 1    LINE 3                   SREC   4
TREC     FILE 4    RECORD 1     LINE 4                  TREC
UREC     FILE 4    RECORD 1      LINE 5                 UREC
```

```
QREC     FILE 4    RECORD 1  LINE 1                     QREC
RREC     FILE 4    RECORD 1   LINE 2                    RREC
SREC     FILE 4    RECORD 1    LINE 3                   SREC   5
TREC     FILE 4    RECORD 1     LINE 4                  TREC
UREC     FILE 4    RECORD 1      LINE 5                 UREC
```

```
AREC     FILE 1    RECORD 1   LINE 1                    AREC
BREC     FILE 1    RECORD 1    LINE 2                   BREC   6
CREC     FILE 1    RECORD 1     LINE 3                  CREC
DREC     FILE 1    RECORD 1      LINE 4                 DREC
```

```
VREC     FILE 4    RECORD 2  LINE 1                     VREC   7
WREC     FILE 4    RECORD 2   LINE 2                    WREC
```

```
NREC     FILE 3    RECORD 1  LINE 1                     NREC
OREC     FILE 3    RECORD 1   LINE 2                    OREC   8
PREC     FILE 3    RECORD 1    LINE 3                   PREC
```

```
KREC     FILE 2    RECORD 2  LINE 1                     KREC
LREC     FILE 2    RECORD 2   LINE 2                    LREC   9
MREC     FILE 2    RECORD 2    LINE 3                   MREC
```

```
EREC     FILE 1    RECORD 2  LINE 1                     EREC
FREC     FILE 1    RECORD 2   LINE 2                    FREC   10
GREC     FILE 1    RECORD 2    LINE 3                   GREC
```

```
VERIFY.  ILLUSTRATION OF VERIFY STAT.  5
USER,DRF2755,DRF.
GET(TESTCOP)
COPY(TESTCOP,DUMMY,VERIFY)
*
SKIPEI(DUMMY)
COPYCR(TESTCOP,DUMMY)
VERIFY(TESTCOP,DUMMY,N,R)
R  7/8/9 END-OF-RECORD
 INFORMATION TO BE ADDED
 TO FILE TESTCOP
Q  6/7/8/9 END-OF-INFORMATION
```

```
     FULL  DAYFILE.  81/10/07. 11.24.38.*11.24.31* PAGE    1

11.24.31.VERIFY.  ILLUSTRATION OF VERIFY STAT.  5
11.24.31.UCCR, 72070,       0.010KCDS.
11.24.32.USER,DRF2755,.
11.24.33.GET(TESTCOP)
11.24.34.COPY(TESTCOP,DUMMY,VERIFY)
11.24.35. EOI ENCOUNTERED.
11.24.35.*  VERIFY,TESTCOP,DUMMY,A,R,L=OUTPUT,N=0.
11.24.36. VERIFY GOOD.
11.24.36.*
11.24.36.SKIPEI(DUMMY)
11.24.36.COPYCR(TESTCOP,DUMMY)
11.24.36. COPY COMPLETE.
11.24.36.VERIFY(TESTCOP,DUMMY,N,R)
11.24.37. VERIFY ERRORS.
11.24.37.UEAD,       0.001KUNS.
11.24.37.UEPF,       0.009KUNS.
11.24.37.UEMS,       0.474KUNS.
11.24.37.UECP,       0.107SECS.
11.24.37.AESR,       1.175UNTS.
11.24.38.$OUT(*/OP=E)
11.24.38. NO FILES PROCESSED.
11.24.38.$DAYFILE(OUTPUT,JT=D)
11.24.51.UCLP, 72,073,      0.192KLNS.
```

VERIFY ERROR LIST. 81/10/07. 11.24.36. PAGE 1

***** 1 EXCESS RECOFD(S) ON DUMMY 1

```
BYE.     ILLUSTRATION OF RENAME - OUT  6
USER,DRF2755,DRF.
GET(TESTCOP)
RENAME(JUNK=TESTCOP)
COPYCF(JUNK,OUTPUT)
OUT.
COPYCF(JUNK,OUTPUT)
*
* WHERE IS THE OUTPUT FROM THE FIRST
* COPYCF COMMAND
*
PURGE,TESTCOP.
Q  6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/01. 11.35.02.*11.34.55* PAGE    1

11.34.55.BYE.     ILLUSTRATION OF RENAME - OUT  6
11.34.55.UCCR, 64020,      0.011KCDS.
11.34.56.USER,DRF2755,.
11.34.56.GET(TESTCOP)
11.34.57.RENAME(JUNK=TESTCOP)
11.34.57.COPYCF(JUNK,OUTPUT)
11.34.58. COPY COMPLETE.
11.34.59.OUT.
11.34.59. 1 FILE PROCESSED.
11.34.59.COPYCF(JUNK,OUTPUT)
11.35.01. COPY COMPLETE.
11.35.01.*
11.35.01.* WHERE IS THE OUTPUT FROM THE FIRST
11.35.01.* COPYCF COMMAND
11.35.01.*
11.35.01.UEAD,       0.001KUNS.
11.35.01.UEPF,       0.009KUNS.
11.35.01.UEMS,       0.102KUNS.
11.35.01.UECP,       0.031SECS.
11.35.01.AESR,       1.043UNTS.
11.35.01.$OUT(*/OP=E)
11.35.01. NO FILES PROCESSED.
11.35.02.$DAYFILE(OUTPUT,JT=D)
11.36.25.UCLP, 64,023,      0.128KLNS.
```

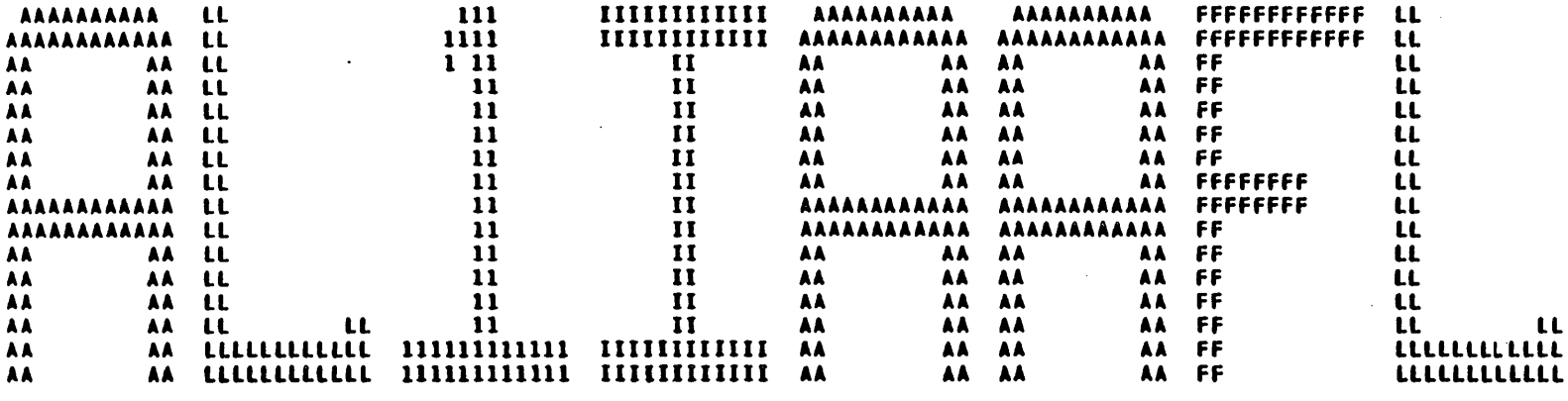
```
OPERATING SYSTEM. = NOS 4F-FNT100        81/09/18. PRINTED = 81/10/01. 11.36.22.

UJN          = BYE       FAMILY      = FEATURE       JOB ORIGIN    = BATCH.
CREATING JSN = AAFI      USER NUMBER = DRF2755        SERVICE CLASS = BATCH.
```

```
 AAAAAAAAAA   LL                111       IIIIIIIIIIII   AAAAAAAAAA    AAAAAAAAAA   FFFFFFFFFFFF  IIIIIIIIIIII
AAAAAAAAAAAA  LL               1111       IIIIIIIIIIII  AAAAAAAAAAAA  AAAAAAAAAAAA  FFFFFFFFFFFF  IIIIIIIIIIII
AA          AA LL              1 11           II        AA          AA AA          AA FF                    II
AA          AA LL                11           II        AA          AA AA          AA FF                    II
AA          AA LL                11           II        AA          AA AA          AA FF                    II
AA          AA LL                11           II        AA          AA AA          AA FF                    II
AA          AA LL                11           II        AA          AA AA          AA FF                    II
AA          AA LL                11           II        AA          AA AA          AA FFFFFFFF              II
AAAAAAAAAAAA  LL                 11           II        AAAAAAAAAAAA  AAAAAAAAAAAA  FFFFFFFF              II
AAAAAAAAAAAA  LL                 11           II        AAAAAAAAAAAA  AAAAAAAAAAAA  FF                    II
AA          AA LL                11           II        AA          AA AA          AA FF                    II
AA          AA LL                11           II        AA          AA AA          AA FF                    II
AA          AA LL                11           II        AA          AA AA          AA FF                    II
AA          AA LL          LL    11           II        AA          AA AA          AA FF                    II
AA          AA LLLLLLLLLLLL  111111111111   IIIIIIIIIIII  AA          AA AA          AA FF                IIIIIIIIIIII
AA          AA LLLLLLLLLLLL  111111111111   IIIIIIIIIIII  AA          AA AA          AA FF                IIIIIIIIIIII
```

```
HREC     FILE 2    RECORD 1  LINE 1                                    HREC
IREC     FILE 2    RECORD 1   LINE 2                                   IREC
JREC     FILE 2    RECORD 1    LINE 3                                  JREC

KREC     FILE 2    RECORD 2  LINE 1                                    KREC
LREC     FILE 2    RECORD 2   LINE 2                                   LREC
MREC     FILE 2    RECORD 2    LINE 3                                  MREC
```

OPERATING SYSTEM = NOS 4F-FNT100 81/09/18. PRINTED = 81/10/01. 11.35.58.

UJN = BYE FAMILY = FEATURE JOB ORIGIN = BATCH.
CREATING JSN = AAFI USER NUMBER = DRF2755 SERVICE CLASS = BATCH.



```
AREC      FILE 1     RECORD 1    LINE 1                    AREC
BREC      FILE 1     RECORD 1     LINE 2                   BREC
CREC      FILE 1     RECORD 1      LINE 3                  CREC
DREC      FILE 1     RECORD 1       LINE 4                 DREC

EREC      FILE 1     RECORD 2    LINE 1                    EREC
FREC      FILE 1     RECORD 2     LINE 2                   FREC
GREC      FILE 1     RECORD 2      LINE 3                  GREC
```

```
          11.36.00.UCLP, 64,023,      0.064KLNS.
```

```
CARRCON. ILLUSTRATE CARRIAGE CONTROL   7
USER,DRF2755,DRF.
*
COPYCR.
BKSP(INPUT)
COPYSBF.
R  7/8/9 END-OF-RECORD
 THIS SINGLE SPACE LINE IS FIRST LINE OF RECORD.  COLUMN 1 IS BLANK.
1THIS PAGE EJECT LINE IS SECOND LINE OF RECORD.  COLUMN 1 IS A ONE.
 SINGLE SPACE LINE.  COLUMN ONE IS BLANK.
0DOUBLE SPACE LINE.  COLUMN ONE IS ZERO.
 SINGLE SPACE.
+              SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
+              SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
+              SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
+              SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
 SINGLE SPACE.
 SINGLE SPACE.
0  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/01. 11.35.00.*11.34.57* PAGE    1

11.34.57.CARRCON. ILLUSTRATE CARRIAGE CONTROL   7
11.34.57.UCCR, 64020,      0.017KCDS.
11.34.57.USER,DRF2755,.
11.34.57.*
11.34.57.COPYCR.
11.34.57. COPY COMPLETE.
11.34.58.BKSP(INPUT)
11.34.58.COPYSBF.
11.34.59. COPY COMPLETE.
11.34.59.UEAD,       0.001KUNS.
11.34.59.UEMS,       0.067KUNS.
11.34.59.UECP,       0.020SECS.
11.34.59.AESR,       1.027UNTS.
11.34.59.$OUT(*/DP=E)
11.35.00. NO FILES PROCESSED.
11.35.00.$DAYFILE(OUTPUT,JT=D)
11.36.04.UCLP, 64,023,      0.256KLNS.
```

1

2

```
THIS SINGLE SPACE LINE IS FIRST LINE OF RECORD.  COLUMN 1 IS BLANK.
```

1

```
THIS PAGE EJECT LINE IS SECOND LINE OF RECORD.  COLUMN 1 IS A ONE.
SINGLE SPACE LINE.  COLUMN ONE IS BLANK.

DOUBLE SPACE LINE.  COLUMN ONE IS ZERO.
SINGLE SPACE. SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
SINGLE SPACE.
SINGLE SPACE.
```



```
 THIS SINGLE SPACE LINE IS FIRST LINE OF RECORD.  COLUMN 1 IS BLANK.
1THIS PAGE EJECT LINE IS SECOND LINE OF RECORD.  COLUMN 1 IS A ONE.
 SINGLE SPACE LINE.  COLUMN ONE IS BLANK.
0DOUBLE SPACE LINE.  COLUMN ONE IS ZERO.
 SINGLE SPACE.
+              SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
+              SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
+              SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
+              SUPPRESS SPACE BEFORE PRINT.  COLUMN ONE IS PLUS.
 SINGLE SPACE.
 SINGLE SPACE.
```

2

# EXERCISE 6

PROBLEM

1. Create a local file containing FORTRAN or COBOL source code. Do a CATALOG and TDUMP on this file. (The source code should contain a main program and at least one subroutine.)

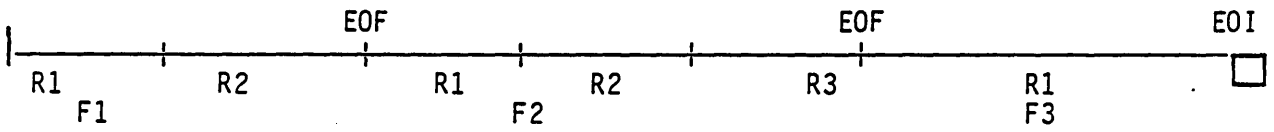
2. Compile the source-code file created in step 1. Do a CATALOG and TDUMP on the relocatable binary file.

3. Use GTR to copy just the subroutine off the binary file. CATALOG the results to check your success.

4. Use COPYX to copy just the main program off the binary file. CATALOG the results to check your success.

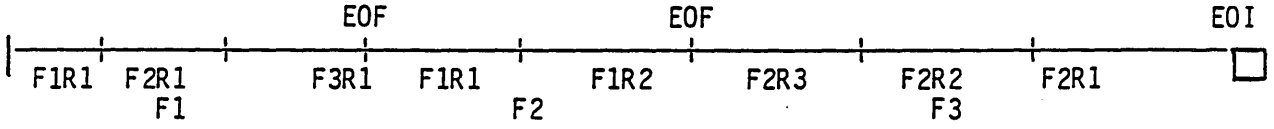
5. Create a file having the indicated structure.

"GIVEN"




6. Use the copy and positioning utilities to create a file with the following structures:

"BUILT"




7. Compare the records in F2 of "GIVEN" file against corresponding records in F3 "BUILT" file.

8. Combine F2 of "GIVEN" file into one logical record.

# EXERCISE 7

PROBLEM

1. Create a coded file containing three logical records. Provide the data for the file in your job deck. Set up carriage control so that the first line of each record will be double spaced. Make all lines on the first record double spaced and all others single spaced when file is printed.

2. Print the file six times, once each using COPYSBF, COPYCF, COPYBF, COPY, COPYCR, and COPYBR.

3. Print only the second record.

4. Create a new file consisting of the same three records in reverse order.

5. Use the ROUTE control card to print this new file.

6. Rename one of the files.

7. Print the renamed file, using the new name.

8. Print the renamed file, using the old name.

9. Copy your entire file INPUT to your file OUTPUT.

# LESSON 4
# FILE PRESERVATION AND PERMANENT FILES

LESSON PREVIEW:

In this lesson, the student will be shown the difference between the two types of permanent files in the system, when it is appropriate to use either, how to create permanent files using various privacy and access mode parameters, how to access permanent files under one's own or another's account number, and how to purge permanent files.

OBJECTIVES:

After completing this lesson the student should be able to -

- State the types of permanent file available and the use for each.
- Describe the permanent file subsystem components and how permanent files are stored and accessed.
- Use the permanent file statements to create, inventory, modify, rename, access and purge permanent files.

TRAINING AIDS:

Visuals V4-1 through V4-22.

PROJECTS:

Exercises 8 and 9 (problems).

REFERENCES:

NOS Reference Manual, Volume 1, Section 8.

## PERMANENT FILE TYPES AND RESIDENCE - OVERVIEW

A permanent file must be explicitly created by the programmer by means of the SAVE or DEFINE control statement. A permanent file must be explicitly deleted using PURGE or PURGALL.

The SAVE statement creates indirect access permanent files. After a SAVE control statement is issued, referencing a local file, there are two copies of the file . . . the original local file, intact and current position unchanged, and the permanent copy made of it on the master device identified by the user number.

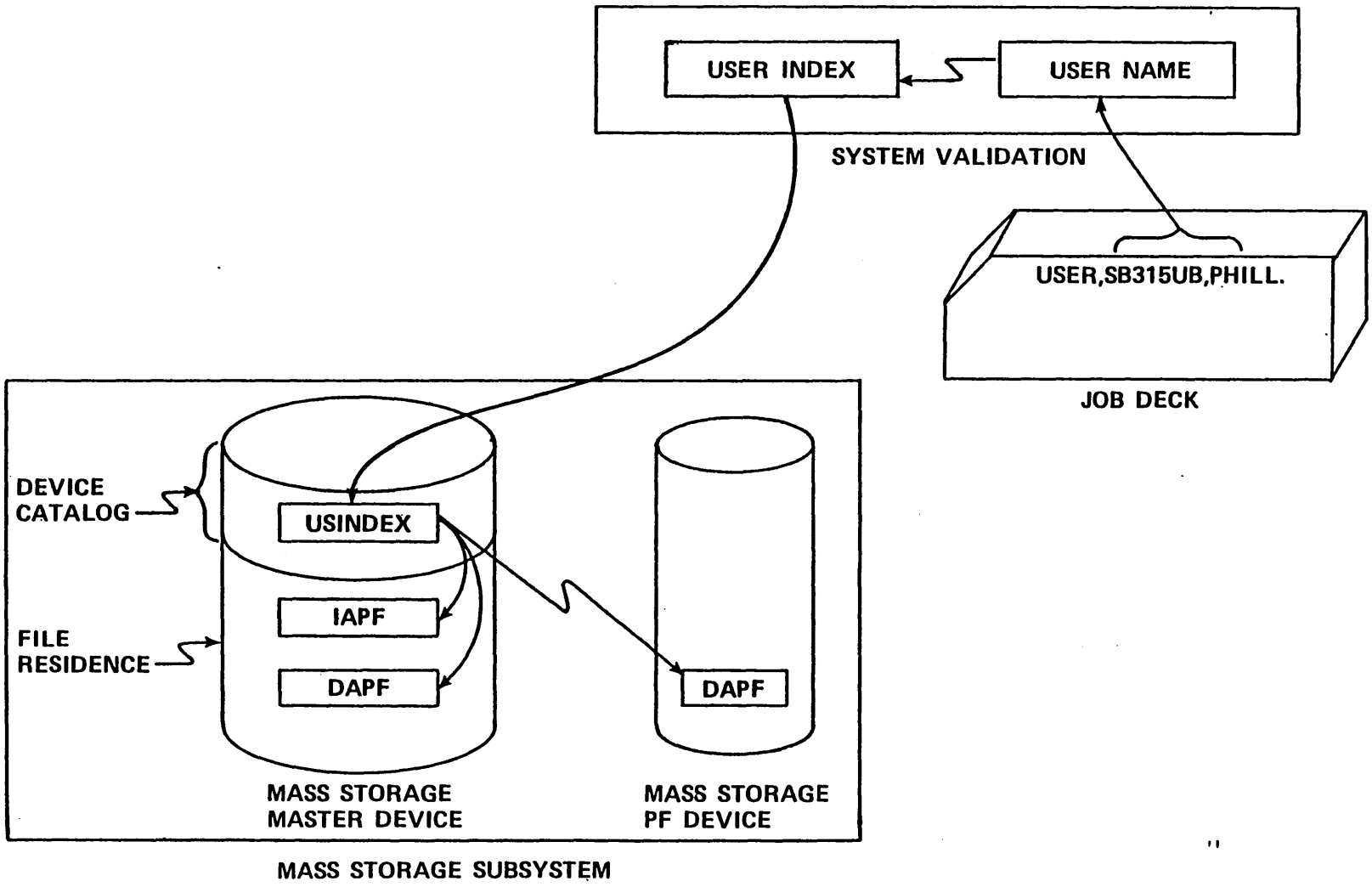
The DEFINE creates direct access permanent files. After a DEFINE, the original file is intact with current position unchanged. It is the permanent file.

For both types of permanent files, indirect and direct access, a catalog entry describing the file physically and logically is made into the catalog section of the master device.

The system associates the user's USER Name with an internal User Index. This User Index identifies the unique master device for a user. All of a given user's indirect access files are on this master device. Direct access files, however, may be resident on the master device or some other device. These relationships are described in visual V4-1.

PURGE or PURGALL, used on either file types, eliminates a file by removing its catalog entry.

## Permanent File Structure and Accounting Elements




LEGEND:

IAPF = INDIRECT ACCESS PERMANENT FILE
DAPF = DIRECT ACCESS PERMANENT FILE

**Permanent File Command Summary**

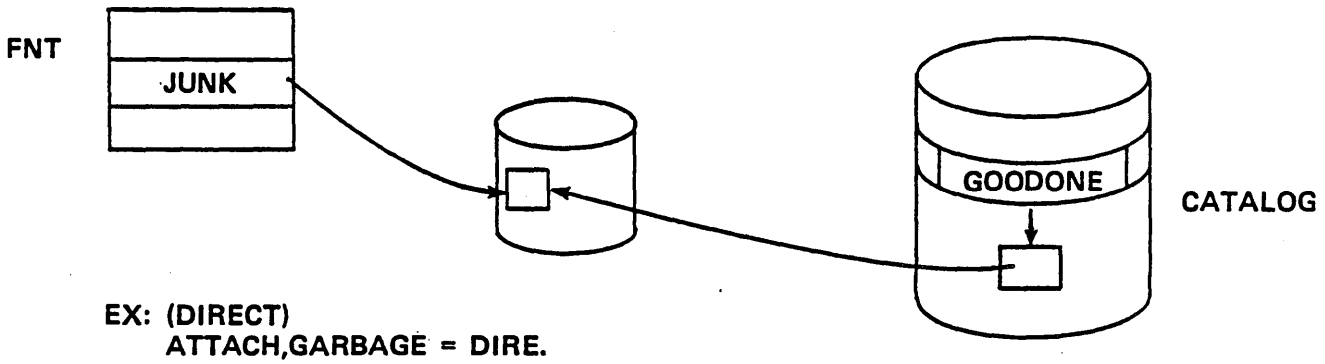
| COMMAND | DIRECT | INDIRECT | |
|---|---|---|---|
| APPEND | | X | |
| ATTACH | X | | ACCESS |
| CATLIST | X | X | |
| CHANGE | X | X | |
| DEFINE | X | | CREATE |
| GET | | X | ACCESS |
| PACKNAM | X | X | |
| PERMIT | X | X | |
| PURGALL | X | X | |
| PURGE | X | X | DELETE |
| REPLACE | | X | ALTER DATA |
| SAVE | | X | CREATE |
| OLD | | X | DECLARE PRIMARY FILE |

GENERAL FORM OF PERMANENT FILE COMMAND

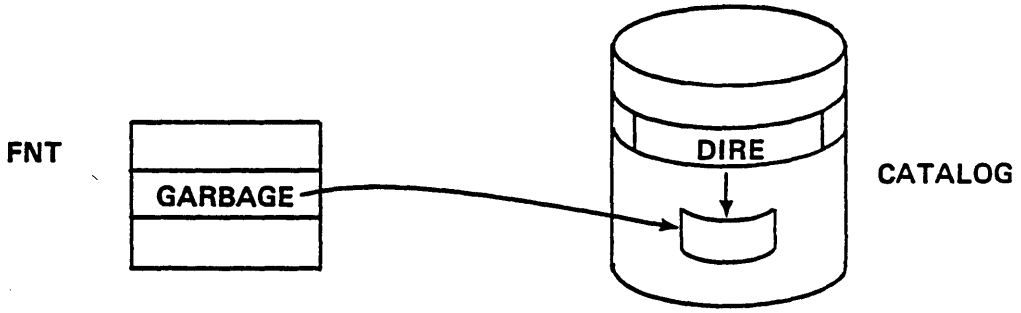
STATEMENT, LFN = PFN.

WHERE
LFN - LOCAL FILE NAME
PFN - PERMANENT FILE NAME

EX: (INDIRECT)
GET, JUNK = GOODONE.




EX: (DIRECT)
ATTACH,GARBAGE = DIRE.



```
CREATPF.   CREATE INDIRECT ACCESS FILE-1
USER,DRF2755,DRF.
*
****************************************
* CREATE INDIRECT ACCESS FILE WITH     *
* MINIMUM PARAMETERS                   *
****************************************
*
****************************************
* CREATE A LOCAL FILE CALLED WORKER    *
****************************************
*
COPYEI(INPUT,WORKER)
*
****************************************
* MAKE IT A PERMANENT FILE             *
****************************************
*
SAVE(WORKER=GOODONE)
*
****************************************
* FIND OUT WHAT PERMANENT FILES PRESENT*
****************************************
*
CATLIST(LO=F,FN=GOODONE)
*
****************************************
* SEE OUTPUT NUMBER 1                  *
****************************************
*
R  7/8/9 END-OF-RECORD
1HERE IS SOME DATA
 TO PUT INTO AN
 INDIRECT ACCESS FILE
 RECORD NUMBER ONE
R  7/8/9 END-OF-RECORD
0IT PROBARLY CONTAINS
 SOURCE CODE
 OR RELOCATEARLE BINARIES
 OR AN ARSOLUTE OVERLAY
 OR AN UPDATE LIBRARY
 OR .......
 RECORD NUMBER TWO
Q  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.18.13.*11.18.11* PAGE    1

11.18.11.CREATPF.   CREATE INDIRECT ACCESS FILE-1
11.18.11.UCOR, 72070,      0.041KCDS.
11.18.11.USER,CRF2755,.
11.18.11.*
11.18.11.****************************************
11.18.11.* CREATE INDIRECT ACCESS FILE WITH     *
11.18.11.* MINIMUM PARAMETERS                   *
11.18.11.****************************************
11.18.11.*
11.18.11.****************************************
11.18.11.* CREATE A LOCAL FILE CALLED WORKER    *
11.18.11.****************************************
11.18.11.*
11.18.11.COPYEI(INPUT,WORKER)
11.18.11. EOI ENCOUNTERED.
11.18.11.*
11.18.11.****************************************
11.18.11.* MAKE IT A PERMANENT FILE             *
11.18.11.****************************************
11.18.11.*
11.18.11.SAVE(WORKER=GOODONE)
11.18.12.*
11.18.12.****************************************
11.18.12.* FIND OUT WHAT PERMANENT FILES PRESENT*
11.18.12.****************************************
11.18.12.*
11.18.12.CATLIST(LO=F,FN=GOODONE)
11.18.12. CATLIST COMPLETE.
11.18.12.*
11.18.12.****************************************
11.18.12.* SEE OUTPUT NUMBER 1                  *
11.18.12.****************************************
11.18.12.*
11.18.12.UEAD,      0.001KUNS.
11.18.12.UEPF,      0.010KUNS.
11.18.12.UEMS,      0.134KUNS.
11.18.12.UECP,      0.022SECS.
11.18.12.AESR,      1.040UNTS.
11.18.12.$OUT(*/OP=F)
11.18.13. NO FILES PROCESSED.
11.18.13.$DAYFILE(OUTPUT,JT=0)
11.18.20.UCLF, 72,073,      0.192KLNS.
```

1

```
CATALOG CF  CRF2755          FM/SYS172  81/10/07. 11.18.12.  PAGE   1

   FILE NAME ACCESS FILE-TYPE LENGTH DN CREATION  ACCESS    DATA MOD
    PASSWORD MC/CNT INDEX  PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
    PR ER RS


   1 GOODONE  IND. PRIVATE         3    81/10/07. 81/10/07. 81/10/07.
                0          WRITE       11.18.12. 11.18.12. 11.18.12.
     N  Y C

        1 INDIRECT ACCESS FILE(S),   TOTAL PRUS =         3.
```

```
REPLACE.  REPLACE INDIRECT ACCESS FILE-2
USER,DRF2755,DRF.
*
****************************************
* REPLACE AN EXISTING PERMANENT FILE   *
* ACCESS THE FILE CREATED IN JOB CREATE*
****************************************
*
GET(GOODONE)
*
****************************************
* PUT NEW INFORMATION IN THE PERM FILE *
****************************************
*
COPYCR(INPUT,GOODONE)
*
****************************************
* LOOK AT THE CHANGE                   *
****************************************
*
REWIND(GOODONE)
COPYEI(GOODONE,OUTPUT)
*
****************************************
* SEE OUTPUT NUMBER 1                  *
* THIS IS NOT WHAT I WANTED TO DO, WILL*
* TRY AGAIN - HAVE TO GET NEW COPY     *
****************************************
*
GET(GOODONE)
BKSP(INPUT,1)
SKIPR(GOODONE,2)
COPYCR(INPUT,GOODONE)
*
****************************************
* CHECK THE FILE JUST CREATED          *
****************************************
*
REWIND(GOODONE)
COPYEI(GOODONE,OUTPUT)
*
****************************************
* SEE OUTPUT NUMBER 2                  *
* THIS IS WHAT I EXPECTED - WE WILL    *
* KEEP THIS COPY AND MAKE IT PERMANENT *
****************************************
*
REPLACE(GOODONE)
*
R  7/8/9 END-OF-RECORD
0HERE IS A RECORD OF
 NEW DATA TO BE PLACED
 IN FILE GOODONE
 IT SHOULD BE THE THIRD RECORD
 RECORD NUMBER THREE
R  7/8/9 END-OF-RECORD
0  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.18.37.*11.18.33* PAGE    1

11.18.33.REPLACE.  REPLACE INDIRECT ACCESS FILE-2
11.18.33.UCCR, 72070,      0.054KCDS.
11.18.33.USER,DRF2755,.
11.18.33.*
11.18.33.****************************************
11.18.33.* REPLACE AN EXISTING PERMANENT FILE   *
11.18.33.* ACCESS THE FILE CREATED IN JOB CREATE*
11.18.33.****************************************
11.18.33.*
11.18.34.GET(GOODONE)
11.18.34.*
11.18.34.****************************************
11.18.34.* PUT NEW INFORMATION IN THE PERM FILE *
11.18.34.****************************************
11.18.34.*
11.18.34.COPYCR(INPUT,GOODONE)
11.18.34. COPY COMPLETE.
11.18.34.*
11.18.34.****************************************
11.18.34.* LOOK AT THE CHANGE                   *
11.18.34.****************************************
11.18.34.*
11.18.34.REWIND(GOODONE)
11.18.34.COPYEI(GOODONE,OUTPUT)
11.18.34. EOI ENCOUNTERED.
11.18.34.*
11.18.34.****************************************
11.18.34.* SEE OUTPUT NUMBER 1                  *
11.18.34.* THIS IS NOT WHAT I WANTED TO DO, WILL*
11.18.34.* TRY AGAIN - HAVE TO GET NEW COPY     *
11.18.34.****************************************
11.18.35.*
11.18.35.GET(GOODONE)
11.18.35.BKSP(INPUT,1)
11.18.35.SKIPR(GOODONE,2)
11.18.35.COPYCR(INPUT,GOODONE)
11.18.35. COPY COMPLETE.
11.18.35.*
11.18.35.****************************************
11.18.35.* CHECK THE FILE JUST CREATED          *
11.18.35.****************************************
11.18.35.*
11.18.35.REWIND(GOODONE)
11.18.36.COPYEI(GOODONE,OUTPUT)
11.18.36. EOI ENCOUNTERED.
11.18.36.*
11.18.36.****************************************
11.18.36.* SEE OUTPUT NUMBER 2                  *
11.18.36.* THIS IS WHAT I EXPECTED - WE WILL    *
11.18.36.* KEEP THIS COPY AND MAKE IT PERMANENT *
11.18.36.****************************************
11.18.36.*
11.18.36.REPLACE(GOODONE)
11.18.36.*
11.18.36.UEAD,      0.001KUNS.
11.18.36.UEPF,      0.015KUNS.
11.18.36.UEMS,      0.276KUNS.
```

HERE IS A RECCRC OF
NEW DATA TO EE PLACED
IN FILE GOODONE
IT SHOULC BE THE THIRD RECORD
RECORD NUMBER THºEE

1

HERE IS SOME DATA
TO PUT INTO AN
INDIRECT ACCESS FILE
RECORD NUMBER ONE

IT PROBABLY CONTAINS
SOURCE CODE
OR RELOCATEABLE BINARIES
OR AN ABSOLUTE OVERLAY
OR AN UPDATE LIBRARY
OR .....
RECORD NUMBER TWO

HERE IS A RECORC OF
NEW DATA TO BE PLACED
IN FILE GOODONE
IT SHOULC BE THE THIRD RECORD
RECORD NUMBER THREE

2

```
APPEND.  ADD INFORMATION TO GOODONE -  3
USER,DRF2755,DRF.
*
****************************************
* ILLUSTRATE APPEND STATEMENT          *
*                                      *
* THIS TIME WE WILL DO THE SAME THING  *
* WE DID IN THE REPLACE JOB - THIS TIME*
* USING THE APPEND STATEMENT           *
*                                      *
* REMEMBER THAT THE FILE HAS 3 RECORDS *
****************************************
*
GET(GOODONE)
COPYCR(INPUT,TEMP)
APPEND(GOODONE,TEMP)
*
****************************************
* CHECK THE RESULTS OF THE APPEND      *
****************************************
*
REWIND(GOODONE)
COPYEI(GOODONE,OUTPUT)
*
****************************************
* CHECK OUTPUT NUMBER 1                *
*                                      *
* THIS IS NOT WHAT I EXPECTED - WHY NOT*
*                                      *
* CHECK THE ORIGIONAL VERSION          *
****************************************
*
GET(GOODONE)
COPYEI(GOODONE,OUTPUT)
*
****************************************
* WHAT HAPPENED HERE SEE OUTPUT NUMB. 2*
****************************************
*
R  7/8/9 END-OF-RECORD
OHERE IS A RECORD
 FOR TRYING OUT APPEND
 WHY NOT DOUBLE SPACED
 LIKE THE PREVIOUS RECORDS
 RECORD NUMBER FOUR
Q  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.20.56 /*11.20.52* PAGE    1

11.20.52.APPEND.  ADD INFORMATION TO GOODONE -  3
11.20.52.UCOR, 72070,      0.044KCDS.
11.20.53.USER,CRF2755,.
11.20.53.*
11.20.53.****************************************
11.20.53.* ILLUSTRATE APPEND STATEMENT          *
11.20.53.*                                      *
11.20.53.* THIS TIME WE WILL DO THE SAME THING  *
11.20.53.* WE DID IN THE REPLACE JOB - THIS TIME*
11.20.53.* USING THE APPEND STATEMENT           *
11.20.53.*                                      *
11.20.53.* REMEMBER THAT THE FILE HAS 3 RECORDS *
11.20.53.****************************************
11.20.53.*
11.20.53.GET(GOODONE)
11.20.53.COPYCR(INPUT,TEMP)
11.20.53. COPY COMPLETE.
11.20.54.APPEND(GOODONE,TEMP)
11.20.54.*
11.20.54.****************************************
11.20.54.* CHECK THE RESULTS OF THE APPEND      *
11.20.54.****************************************
11.20.54.*
11.20.54.REWIND(GOODONE)
11.20.54.COPYEI(GOODONE,OUTPUT)
11.20.54. EOI ENCOUNTERED.
11.20.54.*
11.20.54.****************************************
11.20.54.* CHECK OUTPUT NUMBER 1                *
11.20.54.*                                      *
11.20.54.* THIS IS NOT WHAT I EXPECTED - WHY NOT*
11.20.54.*                                      *
11.20.55.* CHECK THE ORIGIONAL VERSION          *
11.20.55.****************************************
11.20.55.*
11.20.55.GET(GOODONE)
11.20.55.COPYEI(GOODONE,OUTPUT)
11.20.55. EOI ENCOUNTERED.
11.20.55.*
11.20.55.****************************************
11.20.55.* WHAT HAPPENED HERE SEE OUTPUT NUMB. 2*
11.20.55.****************************************
11.20.55.*
11.20.55.UEAD,      0.001KUNS.
11.20.55.UEPF,      0.034KUNS.
11.20.55.UEMS,      0.242KUNS.
11.20.55.UECP,      0.030SECS.
11.20.55.AESR,      1.065UNTS.
11.20.55.$OUT(*/OP=E)
11.20.55. NO FILES PROCESSED.
11.20.56.$DAYFILE(OUTPUT,JT=D)
11.21.05.UCLP, 72,073,      0.256KLNS.
```

```
HERE IS SOME DATA
TO PUT INTO AN
INDIRECT ACCESS FILE
RECORD NUMBER ONE

IT PROBABLY CONTAINS
SOURCE CODE
OR RELOCATEABLE BINARIES
OR AN ABSOLUTE OVERLAY
OR AN UPDATE LIBRARY
OR .......
RECORD NUMBER TWO

HERE IS A RECORD OF
NEW DATA TO BE PLACED
IN FILE GOODONE
IT SHOULD BE THE THIRD RECORD
RECORD NUMBER THREE
```

1



```
HERE IS SOME DATA
TO PUT INTO AN
INDIRECT ACCESS FILE
RECORD NUMBER ONE

IT PROBABLY CONTAINS
SOURCE CODE
OR RELOCATEABLE BINARIES
OR AN ABSOLUTE OVERLAY
OR AN UPDATE LIBRARY
OR .......
RECORD NUMBER TWO

HERE IS A RECORD OF
NEW DATA TO BE PLACED
IN FILE GOODONE
IT SHOULD BE THE THIRD RECORD
RECORD NUMBER THREE
HERE IS A RECORD
FOR TRYING OUT APPEND
WHY NOT DOUBLE SPACED
LIKE THE PREVIOUS RECORDS
RECORD NUMBER FOUR
```

2

```
CREATDA.   CREATE DIRECT ACCESS FILE - 4
USER,DRF2755,DRF.
*
****************************************
* CREATE DIRECT ACCESS FILE WITH       *
* MINIMUM PARAMETERS                   *
*                                      *
* ASK FOR A DIRECT ACCESS FILE         *
****************************************
*
DEFINE(DIRE)
*
****************************************
* CHECK WHAT THE SYSTEM GAVE US        *
****************************************
*
CATLIST(LO=F,FN=DIRE)
*
****************************************
* CHECK OUTPUT NUMBER 1                *
*                                      *
* PLACE SOME INFORMATION INTO THE FILE *
****************************************
*
COPYEI(INPUT,DIRE)
*
****************************************
* CHECK THE FILE AGAIN VIA CATLIST     *
****************************************
*
CATLIST(LO=F,FN=DIRE)
*
****************************************
* CHECK OUTPUT NUMBER 2                *
****************************************
*
R  7/8/9 END-OF-RECORD
1HERE IS A NIFTY RECORD
 FOR MY DIRECT ACCESS FILE
 RECORD NUMBER ONE
R  7/8/9 END-OF-RECORD
0HERE IS ANOTHER ONE - NUMBER TWO
 RECORD NUMBER TWO
R  7/8/9 END-OF-RECORD
0HERE IS NUMBER THREE
 THIS FILE PROBABLY CONTAINS A HUGE
 AMOUNT OF DATA TO BE USED BY SOME
 VERY EXPERIENCED PROGRAMMER WHO HAS
 NOTHING BETTER TO DO THEN ENTER GARBAGE
R  7/8/9 END-OF-RECORD
0THIS DATA CAN GO
 ON AND
 ON AND
 ON AND .........
Q  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.21.17.*11.21.14* PAGE    1

11.21.14.CREATDA.   CREATE DIRECT ACCESS FILE - 4
11.21.14.UCCR, 72070,       0.050KCDS.
11.21.15.USER,DRF2755,.
11.21.15.*
11.21.15.****************************************
11.21.15.* CREATE DIRECT ACCESS FILE WITH       *
11.21.15.* MINIMUM PARAMETERS                   *
11.21.15.*                                      *
11.21.15.* ASK FOR A DIRECT ACCESS FILE         *
11.21.15.****************************************
11.21.15.*
11.21.15.DEFINE(DIRE)
11.21.15.*
11.21.15.****************************************
11.21.15.* CHECK WHAT THE SYSTEM GAVE US        *
11.21.15.****************************************
11.21.15.*
11.21.15.CATLIST(LO=F,FN=DIRE)
11.21.16. CATLIST COMPLETE.
11.21.16.*
11.21.16.****************************************
11.21.16.* CHECK OUTPUT NUMBER 1                *
11.21.16.*                                      *
11.21.16.* PLACE SOME INFORMATION INTO THE FILE *
11.21.16.****************************************
11.21.16.*
11.21.16.COPYEI(INPUT,DIRE)
11.21.16. EOI ENCOUNTERED.
11.21.16.*
11.21.16.****************************************
11.21.16.* CHECK THE FILE AGAIN VIA CATLIST     *
11.21.16.****************************************
11.21.16.*
11.21.16.CATLIST(LO=F,FN=DIRE)
11.21.16. CATLIST COMPLETE.
11.21.16.*
11.21.16.****************************************
11.21.16.* CHECK OUTPUT NUMBER 2                *
11.21.16.****************************************
11.21.16.*
11.21.16.UEAD,       0.001KUNS.
11.21.16.UEPF,       0.018KUNS.
11.21.16.UEMS,       0.182KUNS.
11.21.16.UECP,       0.025SECS.
11.21.16.AESR,       1.049UNTS.
11.21.16.$OUT(*/OP=F)
11.21.17. NO FILES PROCESSED.
11.21.17.$DAYFILE(OUTPUT,JT=D)
11.21.25.UCLP, 72,073,      0.256KLNS.
```

```
CATALOG CF  DRF2755          FM/SYS172  81/10/07. 11.21.16.  PAGE   1

   FILE NAME ACCESS FILE-TYPE LENGTH DN CREATION  ACCESS    DATA MOD
    PASSWORD MO/CNT INDEX  PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
    PR BR RS


   1 DIRE     DIR. PRIVATE          0  * 81/10/07. 81/10/07. 81/10/07.
                 0           WRITE       11.21.15. 11.21.15. 11.21.15.
     N  Y D

        1 DIRECT ACCESS FILE(S),     TOTAL PRUS =         0.
```

1



2

```
CATALOG CF  DRF2755          FM/SYS172  81/10/07. 11.21.16.  PAGE   1

   FILE NAME ACCESS FILE-TYPE LENGTH DN CREATION  ACCESS    DATA MOD
    PASSWORD MO/CNT INDEX  PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME
    PR BR RS


   1 DIRE     DIR. PPIVATE          5  * 81/10/07. 81/10/07. 81/10/07.
                 0           WRITE       11.21.15. 11.21.15. 11.21.15.
     N  Y D

        1 DIRECT ACCESS FILE(S),     TOTAL PRUS =         5.
```

```
UPDIRAF. UPDATE DIRECT ACCESS FILE   - 5
USER,DRF2755,DRF.
*
****************************************
* UPDATE DIRECT ACCESS FILE            *
*                                      *
* ACCESS THE FILE AND COPY IT OUT      *
****************************************
*
ATTACH(DIRE)
COPYEI(DIRE,OUTPUT)
*
****************************************
* CHECK OUTPUT NUMBER 1                *
*                                      *
* NOW COPY ONTO FILE DIRE              *
****************************************
*
COPYCR(INPUT,DIRE)
EXIT.  WHY IS THIS CARD NECESSARY
*
****************************************
* WRITING NOT ALLOWED UNLESS EXPLICITLY*
* REQUESTED ON DIRECT ACCESS FILES     *
*                                      *
* TRY AGAIN                            *
****************************************
*
ATTACH(DIRE/M=WRITE)
SKIPEI(DIRE)
BKSP(INPUT)
COPYCR(INPUT,DIRE)
REWIND(DIRE)
COPYEI(DIRE,OUTPUT)
*
****************************************
* CHECK OUTPUT NUMBER 2                *
****************************************
*
R  7/8/9 END-OF-RECORD
OHERE IS SOME INFO TO BE ADDED TO DIRE.
 WILL IT GET INTO THE FILE OR NOT, PLACE
 YOUR BET AT THE APPROPRIATE WINDOW -
 YOUR INSTRUCTOR WILL HOLD ALL BETS
Q  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.21.35.*11.21.32* PAGE    1

11.21.32.UPDIRAF. UPDATE DIRECT ACCESS FILE   - 5
11.21.32.UCCR, 72070,      0.043KCDS.
11.21.32.USER,CRF2755,.
11.21.32.*
11.21.32.****************************************
11.21.32.* UPDATE DIRECT ACCESS FILE            *
11.21.32.*                                      *
11.21.32.* ACCESS THE FILE AND COPY IT OUT      *
11.21.32.****************************************
11.21.32.*
11.21.32.ATTACH(DIRE)
11.21.33.COPYEI(DIRE,OUTPUT)
11.21.33. EOI ENCOUNTERED.
11.21.33.*
11.21.33.****************************************
11.21.33.* CHECK OUTPUT NUMBER 1                *
11.21.33.*                                      *
11.21.33.* NOW COPY ONTO FILE DIRE              *
11.21.33.****************************************
11.21.33.*
11.21.33.COPYCR(INPUT,DIRE)
11.21.33. WRITE ON READ-ONLY FILE DIRE AT 130.
11.21.33.EXIT.  WHY IS THIS CARD NECESSARY
11.21.33.*
11.21.33.****************************************
11.21.33.* WRITING NOT ALLOWED UNLESS EXPLICITLY*
11.21.33.* REQUESTED ON DIRECT ACCESS FILES     *
11.21.33.*                                      *
11.21.33.* TRY AGAIN                            *
11.21.33.****************************************
11.21.33.*
11.21.33.ATTACH(DIRE/M=WRITE)
11.21.34.SKIPEI(DIRE)
11.21.34.BKSP(INPUT)
11.21.34.COPYCR(INPUT,DIRE)
11.21.34. COPY COMPLETE.
11.21.34.REWIND(DIRE)
11.21.34.COPYEI(DIRE,OUTPUT)
11.21.34. EOI ENCOUNTERED.
11.21.34.*
11.21.34.****************************************
11.21.34.* CHECK OUTPUT NUMBER 2                *
11.21.34.****************************************
11.21.34.*
11.21.35.UEAD,       0.001KUNS.
11.21.35.UEPF,       0.016KUNS.
11.21.35.UEMS,       0.248KUNS.
11.21.35.UECP,       0.035SECS.
11.21.35.AESR,       1.073UNTS.
11.21.35.$OUT(*/OP=E)
11.21.35. NO FILES PROCESSED.
11.21.35.$DAYFILE(OUTPUT,JT=D)
11.21.45.UCLP, 72,073,       0.256KLNS.
```

**1**

```
HERF IS A NIFTY RECORD
FOR MY DIRECT ACCESS FILE
RECORD NUMBER ONE

HERF IS ANOTHER ONE - NUMBER TWO
RECORD NUMBER TWO

HERF IS NUMBER THREE
THIS FILE PROBABLY CONTAINS A HUGE
AMOUNT OF DATA TO BE USED BY SOME
VERY EXPERIENCED PROGRAMMER WHO HAS
NOTHING BETTER TO DO THEN ENTER GARBAGE

THIS DATA CAN GO
ON AND
ON AND
ON AND .........
```



**2**

```
HERE IS A NIFTY RECORD
FOR MY DIRECT ACCESS FILE
RECORD NUMBER ONE

HERE IS ANOTHER ONE - NUMBER TWO
RECORD NUMBER TWO

HERE IS NUMBER THREE
THIS FILE PROBABLY CONTAINS A HUGE
AMOUNT OF DATA TO BE USED BY SOME
VERY EXPERIENCED PROGRAMMER WHO HAS
NOTHING BETTER TO DO THEN ENTER GAREAGE

THIS DATA CAN GO
ON AND
ON AND
ON AND .........

HERE IS SOME INFO TO BE ADDED TO DIRE.
WILL IT GET INTO THE FILE OR NOT, PLACE
YOUR BET AT THE APPROPRIATE WINDOW -
YOUR INSTRUCTOR WILL HOLD ALL BETS
```

```
COPYFIL. CREATE COPIES INDIRECT/DIRECT-6
USER,DRF2755,DRF.
*
****************************************
* THIS JOB WILL CREATE MULTIPLE COPIES *
* OF EXISTING DIRECT AND INDIRECT      *
* ACCESS PERMANENT FILES               *
****************************************
*
GET(A=GOODONE,B=GOODONF,C=GOODONF)
SAVE(A=GOODON1,B=GOODON2,C=GOODON3)
DEFINE(DIRE1,DIRE2,DIRE3)
ATTACH(DIRE)
COPY(DIRE,DIRE1)
REWIND(DIRE)
COPY(DIRE,DIRE2)
REWIND(DIRE)
COPY(DIRE,DIRE3)
*
****************************************
* CHECK WHAT FILES WE HAVE CREATED     *
****************************************
*
CATLIST(LO=F)
*
****************************************
* SEE OUTPUT NUMBER 1                  *
****************************************
*
Q  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.21.56J*11.21.49* PAGE    1

11.21.49.COPYFIL. CREATE COPIES INDIRECT/DIRECT-6
11.21.49.UCCR, 72070,       0.029KCDS.
11.21.50.USER,DRF2755,.
11.21.50.*
11.21.50.****************************************
11.21.50.* THIS JOB WILL CREATE MULTIPLE COPIES *
11.21.50.* OF EXISTING DIRECT AND INDIRECT      *
11.21.50.* ACCESS PERMANENT FILES               *
11.21.50.****************************************
11.21.50.*
11.21.50.GET(A=GOODONE,B=GOODONE,C=GOODONE)
11.21.51.SAVE(A=GOODON1,B=GOODON2,C=GOODON3)
11.21.52.DEFINE(DIRE1,DIRE2,DIRE3)
11.21.53.ATTACH(DIRE)
11.21.53.COPY(DIRE,DIRE1)
11.21.53. EOI ENCOUNTERED.
11.21.53.REWIND(DIRE)
11.21.54.COPY(DIRE,DIRE2)
11.21.54. EOI ENCOUNTERED.
11.21.54.REWIND(DIRE)
11.21.54.COPY(DIRE,DIRE3)
11.21.54. EOI ENCOUNTERED.
11.21.54.*
11.21.54.****************************************
11.21.54.* CHECK WHAT FILES WE HAVE CREATED     *
11.21.54.****************************************
11.21.54.*
11.21.55.CATLIST(LO=F)
11.21.56. CATLIST COMPLETE.
11.21.56.*
11.21.56.****************************************
11.21.56.* SEE OUTPUT NUMBER 1                  *
11.21.56.****************************************
11.21.56.*
11.21.56.UEAD,       0.001KUNS.
11.21.56.UEPF,       0.077KUNS.
11.21.56.UEMS,       0.433KUNS.
11.21.56.UECP,       0.319SECS.
11.21.56.AESR,       1.385UNTS.
11.21.56.$OUT(*/OP=E)
11.21.56. NO FILES PROCESSED.
11.21.56.$DAYFILE(OUTPUT,JT=O)
11.22.18.UCLF, 72,073,      0.448KLNS.
```

CATALOG OF CRF2755 FM/SYS172 81/10/07. 11.21.55. PAGE 1

1

| | FILE NAME | ACCESS | FILE-TYPE | LENGTH | ON | CREATION DATE/TIME | ACCESS DATE/TIME | DATA MOD DATE/TIME |
|---|---|---|---|---|---|---|---|---|
| | PASSWORD | MC/CNT | INDEX | PERM. | SUBSYS | | | |
| | PR BR RS | | | | | | | |
| 1 | TEST | IND. | PRIVATE | 1 | | 81/07/09. | 81/07/09. | 81/07/09. |
| | | 0 | WRITE | | | 07.18.13. | 07.18.13. | 07.18.13. |
| | N Y D | | | | | | | |
| 2 | PAM | IND. | PRIVATE | 1 | | 81/07/10. | 81/07/10. | 81/07/10. |
| | | 2 | WRITE | | | 06.56.46. | 07.08.58. | 07.08.58. |
| | N Y D | | | | | | | |

CATALOG OF CRF2755 FM/SYS172 81/10/07. 11.21.55. PAGE 5

| | FILE NAME | ACCESS | FILE-TYPE | LENGTH | ON | CREATION DATE/TIME | ACCESS DATE/TIME | DATA MOD DATE/TIME |
|---|---|---|---|---|---|---|---|---|
| | PASSWORD | MC/CNT | INDEX | PERM. | SUBSYS | | | |
| | PR BR RS | | | | | | | |
| 53 | DIRE | DIR. | PRIVATE | 6 | * | 81/10/07. | 81/10/07. | 81/10/07. |
| | | 3 | WRITE | | | 11.21.15. | 11.21.53. | 11.21.34. |
| | N Y D | | | | | | | |
| 54 | GOODONE | IND. | PRIVATE | 4 | | 81/10/07. | 81/10/07. | 81/10/07. |
| | | 6 | WRITE | | | 11.18.12. | 11.21.51. | 11.20.54. |
| | N Y D | | | | | | | |
| 55 | GOODON1 | IND. | PRIVATE | 4 | | 81/10/07. | 81/10/07. | 81/10/07. |
| | | 0 | WRITE | | | 11.21.51. | 11.21.51. | 11.21.51. |
| | N Y D | | | | | | | |
| 56 | GOODON2 | IND. | PRIVATE | 4 | | 81/10/07. | 81/10/07. | 81/10/07. |
| | | 0 | WRITE | | | 11.21.51. | 11.21.51. | 11.21.51. |
| | N Y D | | | | | | | |
| 57 | GOODON3 | IND. | PRIVATE | 4 | | 81/10/07. | 81/10/07. | 81/10/07. |
| | | 0 | WRITE | | | 11.21.52. | 11.21.52. | 11.21.52. |
| | N Y D | | | | | | | |
| 58 | DIRE1 | DIR. | PRIVATE | 6 | * | 81/10/07. | 81/10/07. | 81/10/07. |
| | | 0 | WRITE | | | 11.21.52. | 11.21.52. | 11.21.52. |
| | N Y D | | | | | | | |
| 59 | DIRE2 | DIR. | PRIVATE | 6 | * | 81/10/07. | 81/10/07. | 81/10/07. |
| | | 0 | WRITE | | | 11.21.52. | 11.21.52. | 11.21.52. |
| | N Y D | | | | | | | |
| 60 | DIRE3 | DIR. | PRIVATE | 6 | * | 81/10/07. | 81/10/07. | 81/10/07. |
| | | 0 | WRITE | | | 11.21.53. | 11.21.53. | 11.21.53. |
| | N Y D | | | | | | | |

54 INDIRECT ACCESS FILE(S), TOTAL PRUS = 677.
6 DIRECT ACCESS FILE(S), TOTAL PRUS = 322.

```
PURGEM.  REMOVE PERMANENT FILES      - 7
USER,DRF2755,DRF.
*
****************************************
* REMOVE ENTRIES FROM PERM FILE CATALOG*
*                                      *
* HOW MANY FILES ARE THERE AT THIS TIME*
****************************************
*
CATLIST(LO=0)
*
****************************************
* SEE OUTPUT NUMBER 1                  *
****************************************
*
PURGE(DIRE1,DIRE2,DIRE3,GOODON1,GOODON2,GOODON3)
*
****************************************
* DID WE DO ANYTHING                   *
****************************************
*
CATLIST.
*
****************************************
* CHECK OUTPUT NUMBER 2                *
****************************************
*
Q  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.22.14.*11.22.10* PAGE    1

11.22.10.PURGEM.  REMOVE PERMANENT FILES      - 7
11.22.10.UCCR, 72070,      0.027KCDS.
11.22.10.USER,DRF2755,.
11.22.10.*
11.22.10.****************************************
11.22.10.* REMOVE ENTRIES FROM PERM FILE CATALOG*
11.22.10.*                                      *
11.22.10.* HOW MANY FILES ARE THERE AT THIS TIME*
11.22.10.****************************************
11.22.10.*
11.22.10.CATLIST(LO=0)
11.22.11. CATLIST COMPLETE.
11.22.11.*
11.22.11.****************************************
11.22.11.* SEE OUTPUT NUMBER 1                  *
11.22.11.****************************************
11.22.11.*
11.22.11.PURGE(DIRE1,DIRE2,DIRE3,GOODON1,GOODON2,GOODON3)
11.22.12.*
11.22.12.****************************************
11.22.12.* DID WE DO ANYTHING                   *
11.22.12.****************************************
11.22.12.*
11.22.13.CATLIST.
11.22.13. CATLIST COMPLETE.
11.22.13.*
11.22.13.****************************************
11.22.13.* CHECK OUTPUT NUMBER 2                *
11.22.13.****************************************
11.22.13.*
11.22.13.UEAC,      0.001KUNS.
11.22.13.UEPF,      0.055KUNS.
11.22.13.UEMS,      0.106KUNS.
11.22.13.UECP,      0.052SECS.
11.22.13.AESR,      1.071UNTS.
11.22.13.$OUT(*/OP=E)
11.22.13. NO FILES PROCESSED.
11.22.14.$DAYFILE(OUTPUT,JT=0)
11.22.26.UCLP, 72,073,      0.256KLNS.
```

```
CATALOG CF  DRF2755          FM/SYS172  81/10/07. 11.22.11.  PAGE   1

INDIRECT ACCESS FILE(S)

 BUILD      M9001      M9335      M9402      M9522      M9652      M9751
 DEMO       M9286      M9336      M9451      M9551      M9671      M9752
 GOODONE    M9313      M9353      M9452      M9552      M9672      PAM
 GOODON1    M9314      M9354      M9471      M9571      M9701      RUN
 GOODON2    M9315      M9355      M9472      M9591      M9702      STLF
 GOODON3    M9316      M9356      M9501      M9601      M9721      TEST
 MTS9JOB    M9333      M9357      M9502      M9602      M9722      ZZZ
 M9000      M9334      M9401      M9521      M9651

DIRECT ACCESS FILE(S)

 DIRE       DIRE1      DIRE2      DIRE3      QACQACR    STL

       54 INDIRECT ACCESS FILE(S),   TOTAL PRUS =       677.
        6 DIRECT ACCESS FILE(S),     TOTAL PRUS =       322.
```

1



2

```
CATALOG CF  CRF2755          FM/SYS172  81/10/07. 11.22.13.  PAGE   1

INDIRECT ACCESS FILE(S)

 BUILD      M9314      M9354      M9452      M9551      M9652      M9751
 DEMO       M9315      M9355      M9471      M9552      M9671      M9752
 GOODONE    M9316      M9356      M9472      M9571      M9672      PAM
 MTS9JOB    M9333      M9357      M9501      M9591      M9701      RUN
 M9000      M9334      M9401      M9502      M9601      M9702      STLF
 M9001      M9335      M9402      M9521      M9602      M9721      TEST
 M9286      M9336      M9451      M9522      M9651      M9722      ZZZ
 M9313      M9353

DIRECT ACCESS FILE(S)

 DIRE       QACQACR    STL

       51 INDIRECT ACCESS FILE(S),   TOTAL PRUS =       665.
        3 DIRECT ACCESS FILE(S),     TOTAL PRUS =       304.
```

```
PERMIT.  ALLOW ALTERNATE USER ACCESS - 8
USER,ORF2755,ORF.
*
****************************************
* GIVE ANOTHER USER ACCESS TO PRIVATE  *
* FILE THAT WE HAVE CREATED            *
*                                      *
* FIND OUT WHAT PERMISSIONS HAVE BEEN  *
* GRANTED THUS FAR                     *
****************************************
*
CATLIST(LO=P,FN=DIRE)
CATLIST(LO=FP,FN=GOODONE)
*
****************************************
* GIVE ALTERNATE USER ACCESS TO FILES  *
*                                      *
* ALSO CREATE ANOTHER DIR. ACCESS FILE *
****************************************
*
DEFINE(DIRE1)
ATTACH(DIRE)
COPY(DIRE,DIRE1)
PERMIT(DIRE,QUIK172=WRITE)
PERMIT(DIRE1,QUIK172=WRITE)
PERMIT(GOODONE,QUIK172=READ)
CATLIST(LO=P,FN=DIRE)
*
****************************************
* CHECK OUTPUT NUMBER 1                *
****************************************
*
CATLIST(LO=FP,FN=GOODONE)
*
****************************************
* CHECK OUTPUT NUMBER 2                *
****************************************
*
CATLIST(LO=FP,FN=DIRE1)
*
****************************************
* CHECK OUTPUT NUMBER 3                *
*                                      *
* THIS COULD BE DANGEROUS - TAKE CARE  *
****************************************
*
CHANGE(DIRE1,QUIK172=APPEND)
EXIT. (WHY IS THIS CARD NECESSARY.....)
CHANGE(DIRE1/M=APPEND,PW=SECRET)
CATLIST(LO=FP,FN=DIRE1)
*
****************************************
* CHECK OUTPUT NUMBER 4                *
****************************************
*
CHANGE(DIRE/PW=SECRET)
CATLIST.
*
****************************************
* CHECK OUTPUT NUMBER 5                *
****************************************
*
6  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.22.44.*11.22.35* PAGE    1

11.22.35.PERMIT.  ALLOW ALTERNATE USER ACCESS - 8
11.22.35.UCOR, 72070,      0.066KCDS.
11.22.35.USER,ORF2755,.
11.22.35.*
11.22.35.****************************************
11.22.35.* GIVE ANOTHER USER ACCESS TO PRIVATE  *
11.22.35.* FILE THAT WE HAVE CREATED            *
11.22.35.*                                      *
11.22.35.* FIND OUT WHAT PERMISSIONS HAVE BEEN  *
11.22.35.* GRANTED THUS FAR                     *
11.22.35.****************************************
11.22.35.*
11.22.35.CATLIST(LO=P,FN=DIRE)
11.22.36. NO PERMITS.
11.22.36.CATLIST(LO=FP,FN=GOODONE)
11.22.36. NO PERMITS.
11.22.36.*
11.22.36.****************************************
11.22.36.* GIVE ALTERNATE USER ACCESS TO FILES  *
11.22.36.*                                      *
11.22.36.* ALSO CREATE ANOTHER DIR. ACCESS FILE *
11.22.36.****************************************
11.22.36.*
11.22.36.DEFINE(DIRE1)
11.22.37.ATTACH(DIRE)
11.22.37.COPY(DIRE,DIRE1)
11.22.37. EOI ENCOUNTERED.
11.22.38.PERMIT(DIRE,QUIK172=WRITE)
11.22.38.PERMIT(DIRE1,QUIK172=WRITE)
11.22.38.PERMIT(GOODONE,QUIK172=READ)
11.22.39.CATLIST(LO=P,FN=DIRE)
11.22.39. CATLIST COMPLETE.
11.22.39.*
11.22.39.****************************************
11.22.39.* CHECK OUTPUT NUMBER 1                *
11.22.39.****************************************
11.22.39.*
11.22.39.CATLIST(LO=FP,FN=GOODONE)
11.22.40. CATLIST COMPLETE.
11.22.40.*
11.22.40.****************************************
11.22.40.* CHECK OUTPUT NUMBER 2                *
11.22.40.****************************************
11.22.40.*
11.22.40.CATLIST(LO=FP,FN=DIRE1)
11.22.40. CATLIST COMPLETE.
11.22.41.*
11.22.41.****************************************
11.22.41.* CHECK OUTPUT NUMBER 3                *
11.22.41.*                                      *
11.22.41.* THIS COULD BE DANGEROUS - TAKE CARE  *
11.22.41.****************************************
11.22.41.*
11.22.41.CHANGE(DIRE1,QUIK172=APPEND)
```

```
11.22.41. APPEND NOT FOUND.
11.22.41.EXIT. (WHY IS THIS CARD NECESSARY.....)
11.22.41.CHANGE(DIRE1/M=APPEND,PW=)
11.22.42.CATLIST(LO=FP,FN=DIRE1)
11.22.42. CATLIST COMPLETE.
11.22.42.*
11.22.42.*****************************************
11.22.42.* CHECK OUTPUT NUMBER 4                  *
11.22.42.*****************************************
11.22.42.*
11.22.42.CHANGE(DIRE/PW=)
11.22.43.CATLIST.
11.22.43. CATLIST COMPLETE.
11.22.43.*
11.22.43.*****************************************
11.22.43.* CHECK OUTPUT NUMBER 5                  *
11.22.43.*****************************************
11.22.43.*
11.22.43.UEAD,       0.001KUNS.
11.22.43.UEPF,       0.110KUNS.
11.22.43.UEMS,       0.403KUNS.
11.22.43.UECP,       0.078SECS.
11.22.43.AESR,       1.137UNTS.
11.22.43.$OUT(*/OP=F)
11.22.43. NO FILES PROCESSED.
11.22.44.$DAYFILE(OUTPUT,JT=D)
11.22.57.UCLF, 72,073,       0.512KLAS.
```

```
CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.22.39.  PAGE   1
        FILE NAME DIRE
 USER NUMBER(S)

 QUIK172

       1 USER(S)
```

1

```
CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.22.40.  PAGE   1
        FILE NAME GOODONE
USER NUMBER    PERM.  ACCESSES  DATE     TIME

  1. QUIK172   READ*         0 81/10/07. 11.22.39.
```

2



```
CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.22.40.  PAGE   1
        FILE NAME DIRE1
USER NUMBER    PERM.  ACCESSES  DATE     TIME

  1. QUIK172   WRITE*         0 81/10/07. 11.22.38.
```

3

```
CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.22.42.  PAGE   1
        FILE NAME DIRE1
USER NUMBER    PERM.  ACCESSES  DATE     TIME

  1. QUIK172   WRITE*         0 81/10/07. 11.22.38.
```

4

```
CATALOG CF  CRF2755          FM/SYS172  81/10/07. 11.22.43.  PAGE   1

INDIRECT ACCESS FILE(S)

 BUILD     M9314     M9354     M9452     M9551     M9652     M9751
 DEMO      M9315     M9355     M9471     M9552     M9671     M9752
 GOODONE   M9316     M9356     M9472     M9571     M9672     PAM
 MTS9JOB   M9333     M9357     M9501     M9591     M9701     RUN
 M9000     M9334     M9401     M9502     M9601     M9702     STLF
 M9001     M9335     M9402     M9521     M9602     M9721     TEST
 M9286     M9336     M9451     M9522     M9651     M9722     ZZZ
 M9313     M9353

DIRECT ACCESS FILE(S)

 DIRE      DIRE1     QACQACR   STL

      51 INDIRECT ACCESS FILE(S),   TOTAL PRUS =       665.
       4 DIRECT ACCESS FILE(S),     TOTAL PRUS =       310.
```

5

```
ALTACSS. ALTERNATE USER ACCESS TO FILE-9
USER,QUIK172,QUIK.
*
****************************************
* NOTE NEW USER NUMBER BEING USED      *
*                                      *
* GET OTHER PERSONS FILE               *
****************************************
*
GET(GOODONE/UN=DRF2755)
COPYEI(GOODONE,OUTPUT)
*
****************************************
* CHECK OUTPUT NUMBER 1                *
*                                      *
* NOTE THAT READING IS ALLOWED         *
****************************************
*
COPYBR,,GOODONE.
REPLACE(GOODONE/UN=DRF2755)
EXIT.  WHY IS THIS CARD NECESSARY
*
****************************************
* WE ONLY HAVE READ PERMISSION ON FILE *
*                                      *
* TRY ANOTHER FILE                     *
*                                      *
*  WE WILL TRY SOMETHING DRASTIC       *
****************************************
*
PURGE(DIRE1/UN=DRF2755)
EXIT.  WHAT CAUSED THE JOB TO ABORT HERE
PURGE(DIRE1/UN=DRF2755,PW=SECRET)
ATTACH(A=DIRE)
EXIT.    WHY IS THIS CARD NECESSARY
ATTACH(B=DIRE/UN=DRF2755)
EXIT.
ATTACH(C=DIRE/UN=DRF2755,PW=SECRET)
COPY(C)
*
****************************************
* CHECK OUTPUT NUMBER 2                *
****************************************
*
CHANGE(C=DIRE/PW=MINE)
EXIT.    WHY IS THIS CARD NECESSARY
*
*
*
*
REWIND(C)
COPYBR,,C.
EXIT.
ATTACH(C=DIRE/PW=SECRET,UN=DRF2755,M=A)
SKIPEI(C)
BKSP(INPUT,2)
COPYBR(INPUT,C)
REWIND(C)
COPY(C)
*
****************************************
* CHECK OUTPUT NUMBER 3                *
****************************************
```



```
    FULL  DAYFILE.  81/10/07. 11.23.15.*11.23.04* PAGE    1

11.23.04.ALTACSS. ALTERNATE USER ACCESS TO FILE-9
11.23.04.UCOR, 72070,      0.081KCDS.
11.23.04.USER,QUIK172,.
11.23.04.*
11.23.04.****************************************
11.23.04.* NOTE NEW USER NUMBER BEING USED      *
11.23.04.*                                      *
11.23.04.* GET OTHER PERSONS FILE               *
11.23.04.****************************************
11.23.04.*
11.23.04.GET(GOODONE/UN=DRF2755)
11.23.05.COPYEI(GOODONE,OUTPUT)
11.23.05. EOI ENCOUNTERED.
11.23.05.*
11.23.05.****************************************
11.23.05.* CHECK OUTPUT NUMBER 1                *
11.23.05.*                                      *
11.23.05.* NOTE THAT READING IS ALLOWED         *
11.23.05.****************************************
11.23.05.*
11.23.05.COPYBR,,GOODONE.
11.23.05. COPY COMPLETE.
11.23.06.REPLACE(GOODONE/UN=DRF2755)
11.23.06. GOODONE NOT FOUND.
11.23.06.EXIT.  WHY IS THIS CARD NECESSARY
11.23.06.*
11.23.06.****************************************
11.23.06.* WE ONLY HAVE READ PERMISSION ON FILE *
11.23.06.*                                      *
11.23.06.* TRY ANOTHER FILE                     *
11.23.06.*                                      *
11.23.06.*  WE WILL TRY SOMETHING DRASTIC       *
11.23.06.****************************************
11.23.06.*
11.23.06.PURGE(DIRE1/UN=DRF2755)
11.23.07. DIRE1 NOT FOUND.
11.23.07.EXIT.  WHAT CAUSED THE JOB TO ABORT HERE
11.23.07.PURGE(DIRE1/UN=DRF2755,PW=)
11.23.07.ATTACH(A=DIRE)
11.23.08. DIRE NOT FOUND.
11.23.09.EXIT.    WHY IS THIS CARD NECESSARY
11.23.08.ATTACH(B=DIRE/UN=DRF2755)
11.23.09. DIRE NOT FOUND.
11.23.09.EXIT.
11.23.09.ATTACH(C=DIRE/UN=DRF2755,PW=)
11.23.09.COPY(C)
11.23.10. EOI ENCOUNTERED.
11.23.10.*
11.23.10.****************************************
11.23.10.* CHECK OUTPUT NUMBER 2                *
11.23.10.****************************************
11.23.10.*
11.23.10.CHANGE(C=DIRE/PW=)
11.23.10. DIRE NOT FOUND.
11.23.10.EXIT.    WHY IS THIS CARD NECESSARY
11.23.10.*
11.23.10.*
11.23.10.*
```

```
ATTACH(DIRE/UN=DRF2755,PW=SECRET)
EXIT. WHAT CAUSED THE JOB TO ABORT HERE
RETURN(C)
ATTACH(DIRE/UN=DRF2755,PW=SECRET)
COPY(DIRE)
*
****************************************
* CHECK OUTPUT NUMBER 4                *
*                                      *
* PROOF THAT L.F.N.=P.F.N.(DIR.ACCESS) *
*                                      *
* HURRAH WE FINALLY MADE IT            *
****************************************
*
R  7/8/9 END-OF-RECORD
0THIS IS DATA THAT IS
 APPENDED TO DIRECT ACCESS
 FILE DIRE1.................
Q  6/7/8/9 END-OF-INFORMATION
```





```
11.23.10.*
11.23.10.REWIND(C)
11.23.11.COPYER,,C.
11.23.11. WRITE ON READ-ONLY FILE C AT 137.
11.23.11.EXIT.
11.23.11.ATTACH(C=DIRE/PW=,UN=DRF2755,M=A)
11.23.11.SKIPEI(C)
11.23.11.BKSP(INPUT,2)
11.23.12.COPYBR(INPUT,C)
11.23.12. COPY COMPLETE.
11.23.12.REWIND(C)
11.23.12.COPY(C)
11.23.12. EOI ENCOUNTERED.
11.23.12.*
11.23.12.****************************************
11.23.12.* CHECK OUTPUT NUMBER 3                *
11.23.12.****************************************
11.23.12.*
11.23.12.ATTACH(DIRE/UN=DRF2755,PW=)
11.23.13. DIRE BUSY.
11.23.13.EXIT. WHAT CAUSED THE JOB TO ABORT HERE
11.23.13.RETURN(C)
11.23.13.ATTACH(DIRE/UN=DRF2755,PW=)
11.23.14.COPY(DIRE)
11.23.14. EOI ENCOUNTERED.
11.23.14.*
11.23.14.****************************************
11.23.14.* CHECK OUTPUT NUMBER 4                *
11.23.14.*                                      *
11.23.14.* PROOF THAT L.F.N.=P.F.N.(DIR.ACCESS) *
11.23.14.*                                      *
11.23.14.* HURRAH WE FINALLY MADE IT            *
11.23.14.****************************************
11.23.14.*
11.23.14.UEAD,       0.001KUNS.
11.23.14.UEPF,       0.093KUNS.
11.23.14.UEMS,       0.727KUNS.
11.23.14.UECP,       0.082SECS.
11.23.14.AESR,       1.184UNTS.
11.23.15.$OUT(*/OP=E)
11.23.15. NO FILES PROCESSED.
11.23.15.$DAYFILE(OUTPUT,JT=D)
11.23.31.UCLP, 72,073,       0.448KLNS.
```

1

```
HERE IS SOME DATA
TO PUT INTO AN
INDIRECT ACCESS FILE
RECORD NUMBER ONE

IT PROBABLY CONTAINS
SOURCE CODE
OR RELOCATEABLE BINARIES
OR AN ABSOLUTE OVERLAY
OR AN UPDATE LIBRARY
OR .......
RECORD NUMBER TWO

HERE IS A RECORD OF
NEW DATA TO BE PLACED
IN FILE GOODONE
IT SHOULD BE THE THIRD RECORD
RECORD NUMBER THREE
HERE IS A RECORD
FOR TRYING OUT APPEND
WHY NOT DOUBLE SPACED
LIKE THE PREVIOUS RECORDS
RECORD NUMBER FOUR
```



2

```
HERE IS A NIFTY RECORD
FOR MY DIRECT ACCESS FILE
RECORD NUMBER ONE

HERE IS ANOTHER ONE - NUMBER TWO
RECORD NUMBER TWO

HERE IS NUMBER THREE
THIS FILE PROBABLY CONTAINS A HUGE
AMOUNT OF DATA TO BE USED BY SOME
VERY EXPERIENCED PROGRAMMER WHO HAS
NOTHING BETTER TO DO THEN ENTER GARBAGE

THIS DATA CAN GO
ON AND
ON AND
ON AND .........

HERE IS SOME INFO TO BE ADDED TO DIRE.
WILL IT GET INTO THE FILE OR NOT, PLACE
YOUR BET AT THE APPROPRIATE WINDOW -
YOUR INSTRUCTOR WILL HOLD ALL BETS
```

3

```
HERE IS A NIFTY RECORD
FOR MY DIRECT ACCESS FILE
RECORD NUMBER CNE

HERE IS ANOTHER ONE - NUMBER TWO
RECORD NUMBER TWO

HERE IS NUMBER THREE
THIS FILE PROBABLY CONTAINS A HUGE
AMOUNT OF DATA TO BE USED BY SOME
VERY EXPERIENCED PROGRAMMER WHO HAS
NOTHING BETTER TO DO THEN ENTER GARBAGE

THIS DATA CAN GO
ON AND
ON AND
ON AND .........

HERE IS SOME INFO TO BE ADDED TO DIRE.
WILL IT GET INTO THE FILE OR NOT, PLACE
YOUR BET AT THE APPROPRIATE WINDOW -
YOUR INSTRUCTOR WILL HOLD ALL BETS

THIS IS DATA THAT IS
APPENDED TO DIRECT ACCESS
FILE DIRE1...............
```



4

```
HERE IS A NIFTY RECORD
FOR MY DIRECT ACCESS FILE
RECORD NUMBER ONE

HERE IS ANOTHER ONE - NUMBER TWO
RECORD NUMBER TWO

HERE IS NUMBER THREE
THIS FILE PROBABLY CONTAINS A HUGE
AMOUNT OF DATA TO BE USED BY SOME
VERY EXPERIENCED PROGRAMMER WHO HAS
NOTHING BETTER TO DO THEN ENTER GARBAGE

THIS DATA CAN GO
ON AND
ON AND
ON AND .........

HERE IS SOME INFO TO BE ADDED TO DIRE.
WILL IT GET INTO THE FILE OR NOT, PLACE
YOUR BET AT THE APPROPRIATE WINDOW -
YOUR INSTRUCTOR WILL HOLD ALL BETS

THIS IS DATA THAT IS
APPENDED TO DIRECT ACCESS
FILE DIRE1...............
```

```
CHECKEM. CHECK PERMANENT FILES        10
USER,ORF2755,ORF.
*
****************************************
*CHECK FILES STILL IN PERM FILE CATALOG*
****************************************
*
CATLIST(LO=FP,UN=QUIK172,FN=DIRE,NA)
*
****************************************
* CHECK OUTPUT NUMBER 1                *
****************************************
*
CATLIST(UN=QUIK172,FN=GOODONE,NA,LO=FP)
*
****************************************
* CHECK OUTPUT NUMBER 2                *
****************************************
*
CATLIST.
*
****************************************
* CHECK OUTPUT NUMBER 3                *
****************************************
*
PURGE,GOODONE,DIRE.
CATLIST.
*
****************************************
* CHECK OUTPUT NUMBER 4                *
****************************************
*
Q  6/7/8/9 END-OF-INFORMATION
```



```
    FULL  DAYFILE.  81/10/07. 11.23.40.*11.23.36* PAGE    1

11.23.36.CHECKEM. CHECK PERMANENT FILES        10
11.23.36.UCCR, 72070,       0.032KCDS.
11.23.36.USER,ORF2755,,.
11.23.36.*
11.23.36.****************************************
11.23.36.*CHECK FILES STILL IN PERM FILE CATALOG*
11.23.36.****************************************
11.23.36.*
11.23.37.CATLIST(LO=FP,UN=QUIK172,FN=DIRE,NA)
11.23.37. CATLIST COMPLETE.
11.23.37.*
11.23.37.****************************************
11.23.37.* CHECK OUTPUT NUMBER 1                *
11.23.37.****************************************
11.23.37.*
11.23.37.CATLIST(UN=QUIK172,FN=GOODONE,NA,LO=FP)
11.23.38. CATLIST COMPLETE.
11.23.38.*
11.23.38.****************************************
11.23.38.* CHECK OUTPUT NUMBER 2                *
11.23.38.****************************************
11.23.38.*
11.23.38.CATLIST.
11.23.38. CATLIST COMPLETE.
11.23.38.*
11.23.39.****************************************
11.23.39.* CHECK OUTPUT NUMBER 3                *
11.23.39.****************************************
11.23.39.*
11.23.39.PURGE,GOODONE,DIRE.
11.23.39.CATLIST.
11.23.40. CATLIST COMPLETE.
11.23.40.*
11.23.40.****************************************
11.23.40.* CHECK OUTPUT NUMBER 4                *
11.23.40.****************************************
11.23.40.*
11.23.40.UEAD,       0.091KUNS.
11.23.40.UEPF,       0.042KUNS.
11.23.40.UEMS,       0.163KUNS.
11.23.40.UECP,       0.057SECS.
11.23.40.AESR,       1.081UNTS.
11.23.40.$OUT(*/OP=E)
11.23.40. NO FILES PROCESSED.
11.23.40.$DAYFILE(OUTPUT,JT=D)
11.23.50.UCLF, 72,073,      0.384KLNS.
```

```
CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.23.37.  PAGE   1
        FILE NAME DIRE
USER NUMBER    PERM.  ACCESSES  DATE     TIME

  1. QUIK172   WRITE*         3 81/10/07. 11.23.14.
```

1

```
        CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.23.38.  PAGE   1
                FILE NAME GOODONE
        USER NUMBER    PERM.  ACCESSES  DATE     TIME

          1. QUIK172   READ*          1 81/10/07. 11.23.05.
```

2

```
                CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.23.38.  PAGE   1

                INDIRECT ACCESS FILE(S)

                 BUILD     M9314     M9354     M9452     M9551     M9652     M9751
                 DEMO      M9315     M9355     M9471     M9552     M9671     M9752
                 GOODONE   M9316     M9356     M9472     M9571     M9672     PAM
                 MTS9JOB   M9333     M9357     M9501     M9591     M9701     RUN
                 M9000     M9334     M9401     M9502     M9601     M9702     STLF
                 M9001     M9335     M9402     M9521     M9602     M9721     TEST
                 M9286     M9336     M9451     M9522     M9651     M9722     ZZZ
                 M9313     M9353

                DIRECT ACCESS FILE(S)

                 DIRE      QACQACR   STL

                      51 INDIRECT ACCESS FILE(S),   TOTAL PRUS =       665.
                       3 DIRECT ACCESS FILE(S),     TOTAL PRUS =       305.
```

3

```
                  CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.23.40.  PAGE   1

                  INDIRECT ACCESS FILE(S)

                   BUILD     M9315     M9354     M9452     M9551     M9652     M9751
                   DEMO      M9316     M9355     M9471     M9552     M9671     M9752
                   MTS9JOB   M9333     M9356     M9472     M9571     M9672     PAM
                   M9000     M9334     M9357     M9501     M9591     M9701     RUN
                   M9001     M9335     M9401     M9502     M9601     M9702     STLF
                   M9286     M9336     M9402     M9521     M9602     M9721     TEST
                   M9313     M9353     M9451     M9522     M9651     M9722     ZZZ
                   M9314

                  DIRECT ACCESS FILE(S)

                   QACQACR   STL

                        50 INDIRECT ACCESS FILE(S),   TOTAL PRUS =       661.
                         2 DIRECT ACCESS FILE(S),     TOTAL PRUS =       298.
```

4

# EXERCISE 8

PROBLEM

1. (Required) Create a permanent file of the FORTRAN source code for program MAIN and SUB1.

2. (Required) Access the permanent file created in step 1, compile the FTN source, execute the relocatable binary file LGO, and then make a permanent file of LGO.

3. (Required) Access the relocatable binary file created in step 2 and execute it.

4. (Optional) Purge the relocatable binary permanent file and replace the FTN source file with the expanded FTN program group: MAIN SUB1 SUB2 SUB3.

# EXERCISE 9

PROBLEM

1. Create a file by copying 3 records from your file "INPUT" to the file you are creating.

2. "SAVE" it as a private permanent file. If an alternate user No. is available g that alternate user "READ" permission on the file and require a password.

3. "SAVE" it (with a different permanent file name, as a semi-private permanent file). Require a password for alternate user access.

4. Copy the working file to your file "OUTPUT".

5. Get a "short list" of the files in your permanent file catalog.

6. Print the "permit" information for the two permanent files created.

7. Print a full listing of the catalog entry for only one of the permanent files created.

8. Put a comment in your dayfile.

9. Create a working file from a fourth record in your input file.

10. Append it to the first permanent file created and copy the resulting file to OUTPUT.

11. Replace the file in step 10 with the file in step 9 and copy the resulting file to OUTPUT.

12. If an alternate user No. is available, pretend you are that user and access the file created in step 2.

# LESSON 5
# SOURCE PROGRAM LIBRARY MAINTENANCE

LESSON PREVIEW:

The purpose of a source-program library and how it is created, used, and maintained is discussed in this lesson. Also, UPDATE, the utility librarian, is exercised in its various call parameters and source-file directives.

OBJECTIVES:

After completing this lesson the student should be able to -

- List the main features of a source level library.
- Use the UPDATE utility control parameters and directives to create, access, modify, and regenerate a source program library.

TRAINING AIDS:

Visuals V5-1 through V5-32.

PROJECTS:

Exercise 10 (problem).

REFERENCES:

UPDATE Reference Manual (60449900)

## PURPOSE AND OVERVIEW OF A SOURCE PROGRAM LIBRARY

In the section on permanent files, one of the projects involved is working with a file containing a source program. This file, however, has a couple of shortcomings.

The first one is the lack of editing ability on a line-by-line basis, and the other is the large gaps (blank columns) in most free format coding languages . . . these get stored and take up a great deal of space.

The source program librarian, UPDATE, provides a means of overcoming these two limitations as its basic function as well as other extended functions.

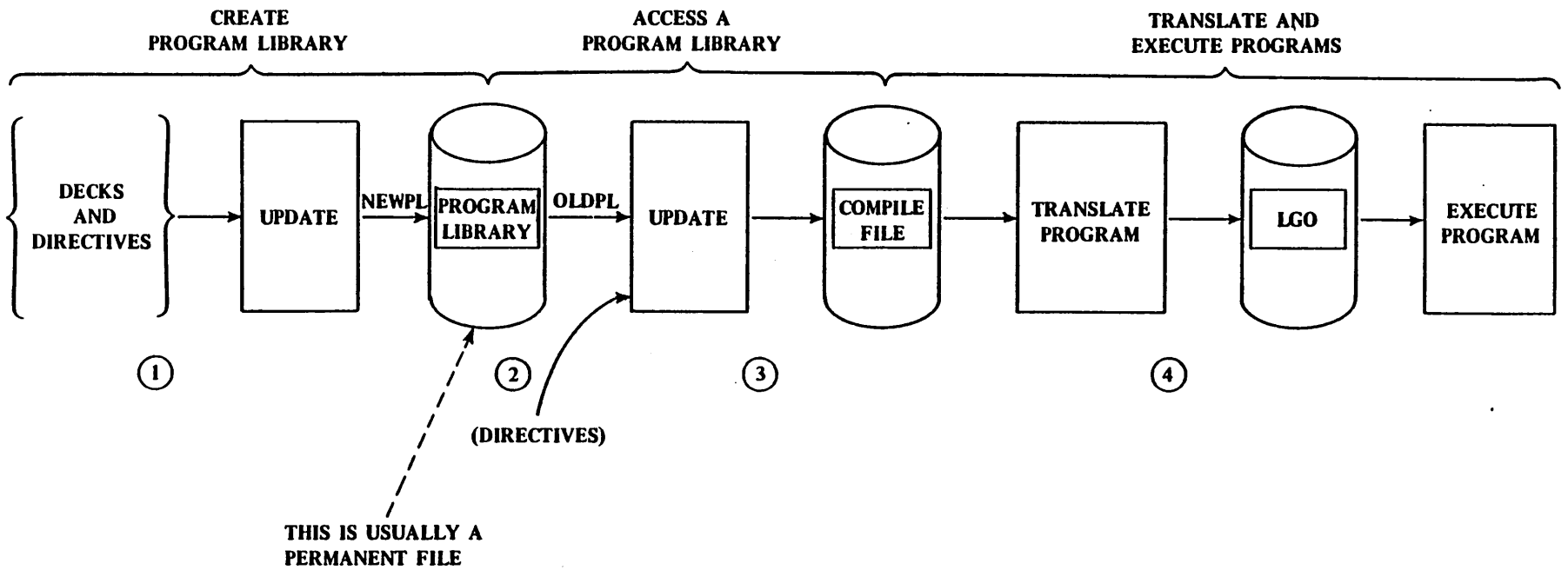
The program library is essentially a group of randomly organized logical records, each of which contains the unit record/card/line image of a deck. Continuing the example of the simple FORTRAN program we have been using, the first card, "PROGRAM MAIN (OUTPUT)", would be the first element of a deck logically named, "MAIN". In order to change it as a later time, it could be referenced as "MAIN.2".

A program library is like any other software . . . it must be created, used and maintained. Visual V5-1 shows the main components and phases of this process. The important files are the program library file and the file on which extracted programs are placed for assembly or compilation.

The program library file when it is first created is named new program library. When it is used subsequently to extract programs from, it is referred to as an old program library.

The file of extracted routines is called the COMPILE file.

The program library can contain any symbolic data, and could more appropriately be referred to as a source program and data library.




(1) THE PROGRAM LIBRARY IS INITIATED AT SOME POINT AS INPUT CARD DECK IMAGES.

(2) THE PROGRAM LIBRARY IS SAVED AS A PERMANENT FILE, IN RANDOM ORGANIZATION FOR PROCESSING EFFICIENCY.

(3) WHEN PROGRAMS ARE REQUIRED, DIRECTIVES EXTRACT DECKS AND WRITE THEM ON A COMPILE FILE.

(4) THE COMPILE FILE IS ASSEMBLED OR COMPILED INTO LOAD MODULES WHICH CAN BE LOADED AND EXECUTED.

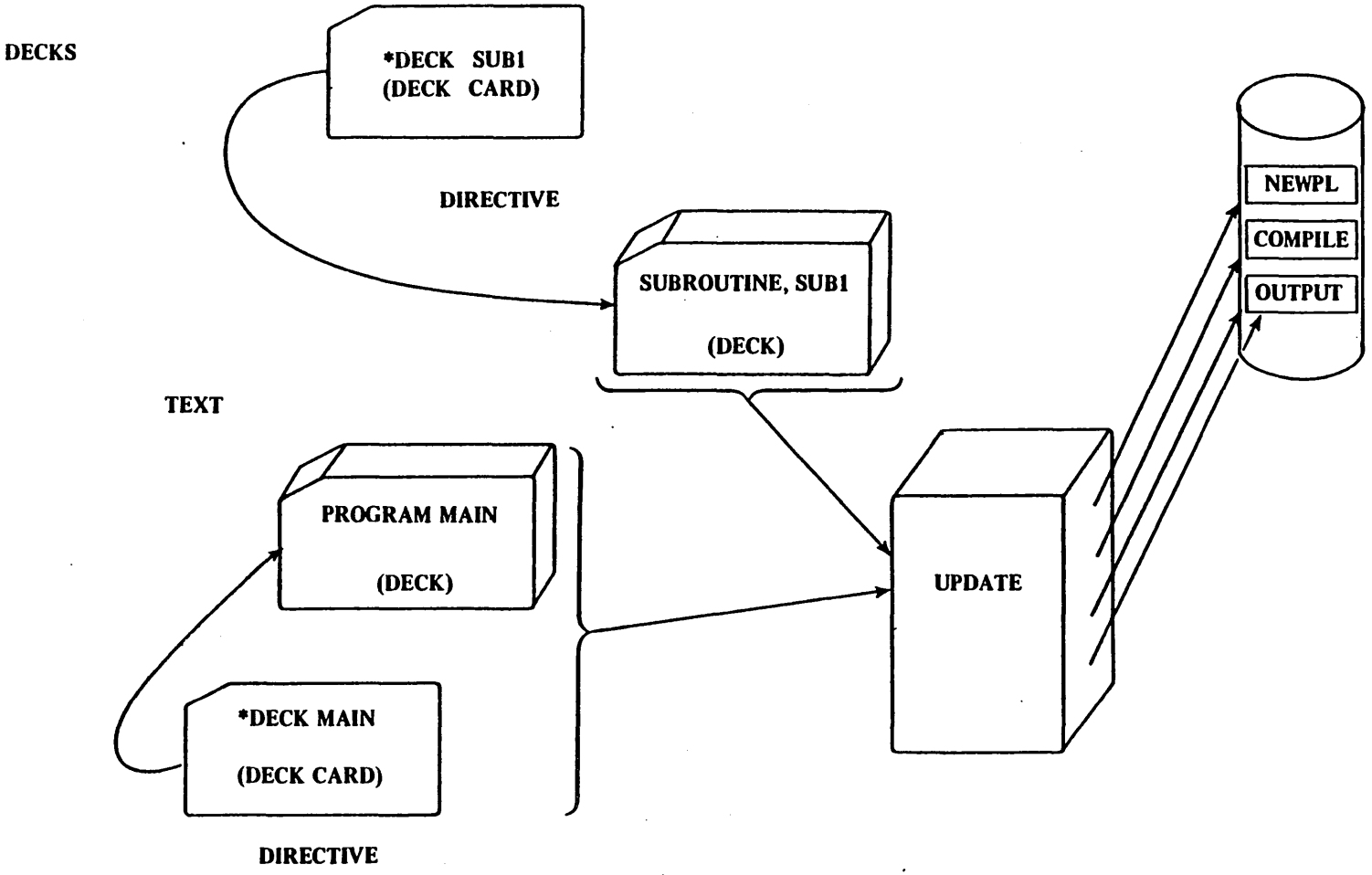
Program Library - Overview of Creation and Access

## CREATION OF A SOURCE PROGRAM LIBRARY

Visual V5-2 shows the elements which are involved in the creation of a source program library. No changes are required for the program decks MAIN and SUB1 other than that the deck name is added as a card in front of each deck. In earlier examples, the FORTRAN programs were on a file as one logical record . . . this provides the convenience of making only one compiler call.

The decks in the current example are one logical record. This is generally true of genuine subroutines. Each deck is therefore kept and referenced as a random record and written with or without modifications to the file from which the compiler can use them. This compile file can be formatted by directives to contain any grouping of logical records or files.

The listing that follows shows the job deck setup for creating a program library which will have two decks on it.




Create a Source Program Library - Overview

## DEFAULTS ON UPDATE CONTROL STATEMENT

P = OLDPL

N = 0 (CORRECTION RUN)  N = NEWPL (CREATION RUN)

S = 0

C = COMPILE

## UPDATE MODES

1. F - FULL UPDATE

   ALL DECKS ON OLD PROGRAM LIBRARY ARE WRITTEN TO NEW PROGRAM LIBRARY, COMPILE FILE, AND SOURCE FILE.

2. Q - QUICK UPDATE

   ONLY DECKS MENTIONED ON COMPILE DIRECTIVES ARE WRITTEN TO COMPILE FILE.

3. NORMAL - SELECTIVE UPDATE

   ONLY DECKS MENTIONED ON COMPILE DIRECTIVES OR DECKS MODIFIED ARE WRITTEN TO THE COMPILE FILE.

```
UPDAT0.
USER,DRF2755,DRF.
UPDATE.
SAVE(NEWPL=DMMPL)
COPYSBF(COMPILE)
REWIND(COMPILE)
FTN,I,R=0,GO.
R  7/8/9 END-OF-RECORD
*DECK ADD
      PROGRAM ADD(OUTPUT,TAPE2=OUTPUT)
C     USE DO STATEMENT TO FORM A LOOP TO ADD 1+2+3+...+100
      ISUM=0
      DO 10 I=1,100
   10 ISUM=ISUM+1
      WRITE(2,*) ISUM
      END
0  6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/01. 11.33.32.*11.32.08* PAGE    1

11.32.08.UPDAT0.
11.32.08.UCCR, 64020,      0.015KCDS.
11.32.08.USER,DRF2755,.
11.32.08.UPDATE.                    1
11.32.09.UPDATE CREATION RUN
11.32.09.CREATING NEW PROGRAM LIBRARY
11.32.09. UPDATE COMPLETE.
11.32.09.SAVE(NEWPL=DMMPL)
11.32.10.COPYSBF(COMPILE)           2
11.32.10. ECI ENCOUNTERED.
11.32.10.REWIND(COMPILE)
11.32.10.FTN,I,R=0,GO.
11.32.13.       .084 CP SECONDS COMPILATION TIME  3
11.32.13.LGO
11.32.16.      END  ADD
11.32.16.     012700 MAXIMUM EXECUTION FL.
11.32.16.       .042 CP SECONDS EXECUTION TIME.
11.32.16.UEAD,       0.001KUNS.
11.32.16.UEPF,       0.005KUNS.
11.32.16.UEMS,       1.395KUNS.
11.32.16.UECP,       0.444SECS.
11.32.16.AESR,       1.766UNTS.
11.32.16.$OUT(*/OP=E)
11.32.16. NO FILES PROCESSED.
11.33.32.$DAYFILE(OUTPUT,JT=D)
11.35.04.UCLP, 64,023,      0.320KLNS.
```

```
CREATION RUN                CARDS ENCOUNTERED IN INPUT          UPDATE 1.4-528.      81/10/01. 11.32.09.      PAGE   1
*****    *DECK ADD                                                                    ADD              1
```

1

```
               CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

         ADD


DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

         YANK$$$    ADD


         DECKS WRITTEN TO COMPILE FILE

         ADD


 THIS UPDATE REQUIRED  37000B WORDS OF CORE.
```



2

```
      PROGRAM ADD(OUTPUT,TAPE2=OUTPUT)                                  ADD        2
C     USE DO STATEMENT TO FORM A LOOP TO ADD 1+2+3+...+100              ADD        3
      ISUM=0                                                            ADD        4
      DO 10 I=1,100                                                     ADD        5
   10 ISUM=ISUM+1                                                       ADD        6
      WRITE(2,*) ISUM                                                   ADD        7
      END                                                               ADD        8
```

3

```
      PROGRAM ADD       73/74   OPT=1                    FTN 4.8+528      81/10/01. 11.32.10      PAGE     1

      1                 PROGRAM ADD(OUTPUT,TAPE2=OUTPUT)                                  ADD        2
                  C     USE DO STATEMENT TO FORM A LOOP TO ADD 1+2+3+...+100              ADD        3
                        ISUM=0                                                            ADD        4
                        DO 10 I=1,100                                                     ADD        5
      5           10    ISUM=ISUM+1                                                       ADD        6
                        WRITE(2,*) ISUM                                                   ADD        7
                        END
```

```
UPDAT1.
USER,DRF2755,DRF.
GET(OLDPL=DMMPL)
UPDATE(S)
COPYSBF(SOURCE)
COPYSBF(COMPILE)
REWIND(COMPILE)
FTN(I,R=0,GO)
PURGE(DMMPL)
R  7/8/9 END-OF-RECORD
*IDENT CORRECT
*INSERT ADD.7
      WRITE(2,*) " THIS OUTPUT COMES FROM THE MODIFIED JOB"
*DELETE ADD.6
   10 ISUM=ISUM+I
Q  6/7/8/9 END-OF-INFORMATION
```

```
 FULL  DAYFILE.  81/10/01. 11.36.21.*11.36.15* PAGE    1

11.36.15.UPDAT1.
11.36.15.UCCR, 64020,      0.014KCDS.
11.36.15.USER,DRF2755,.
11.36.15.GET(OLDPL=DMMPL)
11.36.16.UPDATE(S)                    1
11.36.16. UPDATE COMPLETE.
11.36.16.COPYSBF(SOURCE)
11.36.17. EOI ENCOUNTERED.            2
11.36.17.COPYSBF(COMPILE)                3
11.36.17. EOI ENCOUNTERED.
11.36.17.REWIND(COMPILE)
11.36.17.FTN(I,R=0,GO)
11.36.19.      .075 CP SECONDS COMPILATION TIME  4
11.36.19.LGO
11.36.20.     END  ADD
11.36.20.    012700 MAXIMUM EXECUTION FL.
11.36.20.      .037 CP SECONDS EXECUTION TIME.
11.36.20.PURGE(DMMPL)
11.36.20.UEAD,       0.001KUNS.
11.36.20.UEPF,       0.010KUNS.
11.36.20.UEMS,       1.461KUNS.
11.36.20.UECP,       0.422SECS.
11.36.20.AESR,       1.753UNTS.
11.36.20.$OUT(*/OP=E)
11.36.20. NO FILES PROCESSED.
11.36.21.$DAYFILE(OUTPUT,JT=D)
11.37.06.UCLP, 64,023,      0.384KLNS.
```

```
*****    *IDENT CORRECT
*****    *INSERT ADD.7
               WRITE(2,*) # THIS OUTPUT COMES FROM THE MODIFIED JOB#
*****    *DELETE ADD.6
            10 ISUM=ISUM+I
```

1

MODIFICATIONS / CONTROL CARDS

```
ADD            10 ISUM=ISUM+1                                          ADD          6       D
ADD            10 ISUM=ISUM+I                                          CORRECT      1       I
ADD               WRITE(2,*) # THIS OUTPUT COMES FROM THE MODIFIED JOB# CORRECT      2       I
```

CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

ADD CORRECT

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANK$$$ ADD

DECKS WRITTEN TO COMPILE FILE

ADD

THIS UPDATE REQUIRED 37000B WORDS OF CORE.

**2**

```
*DECK ADD
      PROGRAM ADD(OUTPUT,TAPE2=OUTPUT)
C     USE DO STATEMENT TO FORM A LOOP TO ADD 1+2+3+...+100
      ISUM=0
      DO 10 I=1,100
   10 ISUM=ISUM+I
      WRITE(2,*) ISUM
      WRITE(2,*) # THIS OUTPUT COMES FROM THE MODIFIED JOB#
      END
```

**3**

```
      PROGRAM ADD(OUTPUT,TAPE2=OUTPUT)                          ADD         2
C     USE DO STATEMENT TO FORM A LOOP TO ADD 1+2+3+...+100      ADD         3
      ISUM=0                                                    ADD         4
      DO 10 I=1,100                                             ADD         5
   10 ISUM=ISUM+I                                               CORRECT     1
      WRITE(2,*) ISUM                                           ADD         7
      WRITE(2,*) # THIS OUTPUT COMES FROM THE MODIFIED JOB#     CORRECT     2
      END                                                       ADD         8
```

**4**

```
      PROGRAM ADD        73/74   OPT=1                FTN 4.8+528      81/10/01. 11.36.17      PAGE     1

          1                PROGRAM ADD(OUTPUT,TAPE2=OUTPUT)                          ADD         2
                       C   USE DO STATEMENT TO FORM A LOOP TO ADD 1+2+3+...+100      ADD         3
                           ISUM=0                                                    ADD         4
                           DO 10 I=1,100                                             ADD         5
          5             10 ISUM=ISUM+I                                               CORRECT     1
                           WRITE(2,*) ISUM                                           ADD         7
                           WRITE(2,*) # THIS OUTPUT COMES FROM THE MODIFIED JOB#     CORRECT     2
                           END                                                       ADD         8
     5050
     # THIS OUTPUT COMES FROM THE MODIFIED JOB#
```

```
UPDAT2.
USER,DRF2755,DRF.
UPDATF.
SAVE(NEWPL=DATAPL)
COPYSBF(COMPILE)
R  7/8/9 END-OF-RECORD
*COMDECK COM1
          COMMON DECK 1, CARD 1
          COMMON DECK 1, CARD 2
          COMMON DECK 1, CARD 3
*COMDECK COM2
          COMMON DECK 2, CARD 1
          COMMON DECK 2, CARD 2
          COMMON DECK 2, CARD 3
          COMMON DECK 2, CARD 4
          COMMON DECK 2, CARD 5
*DECK DECK1
**********DECK 1, CARD 1**********
**********DECK 1, CARD 2**********
**********DECK 1, CARD 3**********
**********DECK 1, CARD 4**********
**********DECK 1, CARD 5**********
**********DECK 1, CARD 6**********
*DECK DECK2
**********DECK 2, CARD 1**********
**********DECK 2, CARD 2**********
*CALL COM2
**********DECK 2, CARD 3**********
**********DECK 2, CARD 4**********
**********DECK 2, CARD 5**********
*DECK DECK3
**********DECK 3, CARD 1**********
**********DECK 3, CARD 2**********
*CALL COM1
*CALL COM2
**********DECK 3, CARD 3**********
**********DECK 3, CARD 4**********
**********DECK 3, CARD 5**********
**********DECK 3, CARD 6**********
**********DECK 3, CARD 7**********
**********DECK 3, CARD 8**********
**********DECK 3, CARD 9**********
0  6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/01. 11.38.27.*11.38.23* PAGE    1

11.38.23.UPDAT2.
11.38.23.UCCR, 64020,      0.041KCDS.
11.38.23.USER,DRF2755,.
11.38.24.UPDATE.                    1
11.38.24.UPDATE CREATION RUN
11.38.24.CREATING NEW PROGRAM LIBRARY
11.38.26. UPDATE COMPLETE.
11.38.26.SAVE(NEWPL=DATAPL)
11.38.27.COPYSBF(COMPILE)           2
11.38.27. EOI ENCOUNTERED.
11.38.27.UEAD,       0.001KUNS.
11.38.27.UEPF,       0.009KUNS.
11.38.27.UEMS,       0.361KUNS.
11.38.27.UECP,       0.104SECS.
11.38.27.AESR,       1.176UNTS.
11.38.27.$OUT(*/OP=E)
11.38.27. NO FILES PROCESSED.
11.38.27.$DAYFILE(OUTPUT,JT=D)
11.38.34.UCLP, 64,023,      0.256KLNS.
```

```
*****     *COMDECK COM1
*****     *COMDECK COM2
*****     *DECK DECK1
*****     *DECK DECK2
*****     *CALL COM2
*****     *DECK DECK3
*****     *CALL COM1
*****     *CALL COM2
```

1

```
COM1        1
COM2        1
DECK1       1
DECK2       1
DECK2       4
DECK3       1
DECK3       4
DECK3       5
```

CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

COM1 COM2 DECK1 DECK2 DECK3

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANK$$$ COM1 COM2 DECK1 DECK2 DECK3

2

COMMON DECKS ENCOUNTERED

COM1 COM2

DECKS WRITTEN TO COMPILE FILE

DECK1 DECK2 DECK3

THIS UPDATE REQUIRED 370000B WORDS OF CORE.

```
**********DECK 1, CARD 1**********                DECK1       2
**********DECK 1, CARD 2**********                DECK1       3
**********DECK 1, CARD 3**********                DECK1       4
**********DECK 1, CARD 4**********                DECK1       5
**********DECK 1, CARD 5**********                DECK1       6
**********DECK 1, CARD 6**********                DECK1       7
**********DECK 2, CARD 1**********                DECK2       2
**********DECK 2, CARD 2**********                DECK2       3
         COMMON DECK 2, CARD 1                    COM2        2
         COMMON DECK 2, CARD 2                    COM2        3
         COMMON DECK 2, CARD 3                    COM2        4
         COMMON DECK 2, CARD 4                    COM2        5
         COMMON DECK 2, CARD 5                    COM2        6
**********DECK 2, CARD 3**********                DECK2       5
**********DECK 2, CARD 4**********                DECK2       6
**********DECK 2, CARD 5**********                DECK2       7
**********DECK 3, CARD 1**********                DECK3       2
**********DECK 3, CARD 2**********                DECK3       3
         COMMON DECK 1, CARD 1                    COM1        2
         COMMON DECK 1, CARD 2                    COM1        3
         COMMON DECK 1, CARD 3                    COM1        4
         COMMON DECK 2, CARD 1                    COM2        2
         COMMON DECK 2, CARD 2                    COM2        3
         COMMON DECK 2, CARD 3                    COM2        4
         COMMON DECK 2, CARD 4                    COM2        5
         COMMON DECK 2, CARD 5                    COM2        6
**********DECK 3, CARD 3**********                DECK3       6
**********DECK 3, CARD 4**********                DECK3       7
**********DECK 3, CARD 5**********                DECK3       8
**********DECK 3, CARD 6**********                DECK3       9
**********DECK 3, CARD 7**********                DECK3      10
**********DECK 3, CARD 8**********                DECK3      11
**********DECK 3, CARD 9**********                DECK3      12
```

```
UPDAT3.
USER,DRF2755,DRF.
GET(OLDPL=DATAPL)
UPDATE(N,S)
SAVE(NEWPL=DATAPL1)
COPYSBF(COMPILE)
COPYSBF(SOURCE)
PURGE(DATAPL)
R  7/8/9 END-OF-RECORD
*ADDFILE,INPUT,DECK3
*DECK DECK4
$$$$$$$$$$DECK 4, CARD 1$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 2$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 3$$$$$$$$$$
0  6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/01. 11.39.37.*11.39.28* PAGE    1

11.39.28.UPDAT3.
11.39.28.UCCR, 64020,      0.013KCDS.
11.39.28.USER,DRF2755,.
11.39.28.GET(OLDPL=DATAPL)
11.39.29.UPDATE(N,S)                    1
11.39.30.DECK STRUCTURE CHANGED
11.39.31. UPDATE COMPLETE.
11.39.31.SAVE(NEWPL=DATAPL1)
11.39.33.COPYSBF(COMPILE)               2
11.39.33. EOI ENCOUNTERED.
11.39.34.COPYSBF(SOURCE)
11.39.35. EOI ENCOUNTERED.              3
11.39.36.PURGE(DATAPL)
11.39.36.UEAD,       0.001KUNS.
11.39.36.UEPF,       0.023KUNS.
11.39.36.UEMS,       0.453KUNS.
11.39.36.UECP,       0.119SECS.
11.39.36.AESR,       1.210UNTS.
11.39.36.$OUT(*/OP=E)
11.39.36. NO FILES PROCESSED.
11.39.37.$DAYFILE(OUTPUT,JT=D)
11.39.45.UCLP, 64,023,      0.320KLNS.
```

```
*****    *ADDFILE,INPUT,DECK3
*****    *DECK DECK4                                                    INPUT

         MODIFICATIONS / CONTROL CARDS


DECK4    *DECK DECK4                                          DECK4      1     I
DECK4    $$$$$$$$$$DECK 4, CARD 1$$$$$$$$$$                   DECK4      2     I
DECK4    $$$$$$$$$$DECK 4, CARD 2$$$$$$$$$$                   DECK4      3     I
DECK4    $$$$$$$$$$DECK 4, CARD 3$$$$$$$$$$                   DECK4      4     I


               CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

         COM1      COM2      DECK1     DECK2     DECK3     DECK4


DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

         YANK$$$   COM1      COM2      DECK1     DECK2     DECK3     DECK4


          COMMON DECKS ENCOUNTERED

         COM1      COM2


         DECKS WRITTEN TO COMPILE FILE

         DECK4


 THIS UPDATE REQUIRED  370008 WORDS OF CORE.
```

```
$$$$$$$$$$DECK 4, CARD 1$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 2$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 3$$$$$$$$$$
```

2

```
DECK4       2
DECK4       3
DECK4       4
```

3

```
*COMDECK COM1
         COMMON DECK 1, CARD 1
         COMMON DECK 1, CARD 2
         COMMON DECK 1, CARD 3
*COMDECK COM2
         COMMON DECK 2, CARD 1
         COMMON DECK 2, CARD 2
         COMMON DECK 2, CARD 3
         COMMON DECK 2, CARD 4
         COMMON DECK 2, CARD 5
*DECK DECK1
**********DECK 1, CARD 1**********
**********DECK 1, CARD 2**********
**********DECK 1, CARD 3**********
**********DECK 1, CARD 4**********
**********DECK 1, CARD 5**********
**********DECK 1, CARD 6**********
*DECK DECK2
**********DECK 2, CARD 1**********
**********DECK 2, CARD 2**********
*CALL COM2
**********DECK 2, CARD 3**********
**********DECK 2, CARD 4**********
**********DECK 2, CARD 5**********
*DECK DECK3
**********DECK 3, CARD 1**********
**********DECK 3, CARD 2**********
*CALL COM1
*CALL COM2
**********DECK 3, CARD 3**********
**********DECK 3, CARD 4**********
**********DECK 3, CARD 5**********
**********DECK 3, CARD 6**********
**********DECK 3, CARD 7**********
**********DECK 3, CARD 8**********
**********DECK 3, CARD 9**********
*DECK DECK4
$$$$$$$$$$DECK 4, CARD 1$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 2$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 3$$$$$$$$$$
```

```
UPDAT4.
USER,DRF2755,DRF.
GET(OLDPL=DATAPL1)
UPDATE(F,N,S)
SAVE(NEWPL=DATAPL2)
COPYSBF(COMPILE)
COPYSBF(SOURCE)
R  7/8/9 END-OF-RECORD
*IDENT CORA
*YANKDECK DECK1
*PURDECK DECK2
Q  6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/01. 11.43.16.*11.43.08* PAGE    1

11.43.08.UPDAT4.
11.43.08.UCCR, 64020,       0.010KCDS.
11.43.09.USER,DRF2755,.
11.43.09.GET(OLDPL=DATAPL1)
11.43.10.UPDATE(F,N,S)                    1
11.43.12.DECK STRUCTURE CHANGED
11.43.14. UPDATE COMPLETE.
11.43.14.SAVE(NEWPL=DATAPL2)
11.43.15.COPYSBF(COMPILE)                 2
11.43.15. EOI ENCOUNTERED.
11.43.16.COPYSBF(SOURCE)
11.43.16. EOI ENCOUNTERED.                3
11.43.16.UREAD,       0.001KUNS.
11.43.16.UEPF,        0.018KUNS.
11.43.16.UEMS,        0.431KUNS.
11.43.16.UECP,        0.126SECS.
11.43.16.AESR,        1.215UNTS.
11.43.16.$OUT(*/OP=E)
11.43.16. NO FILES PROCESSED.
11.43.16.$DAYFILE(OUTPUT,JT=D)
11.43.26.UCLP, 64,023,      0.384KLNS.
```

1

```
*****    *IDENT CORA
*****    *YANKDECK DECK1
*****    *PURDECK DECK2
```

MODIFICATIONS / CONTROL CARDS

```
YANK$$$  *YANKDECK DECK1                                          CORA      1      I

DECK1    *DECK DECK1                                              DECK1     1      D
DECK1    **********DECK 1, CARD 1**********                       DECK1     2      D
DECK1    **********DECK 1, CARD 2**********                       DECK1     3      D
DECK1    **********DECK 1, CARD 3**********                       DECK1     4      D
DECK1    **********DECK 1, CARD 4**********                       DECK1     5      D
DECK1    **********DECK 1, CARD 5**********                       DECK1     6      D
DECK1    **********DECK 1, CARD 6**********                       DECK1     7      D

DECK2    *DECK DECK2                                              DECK2     1      P  ACTIVE
DECK2    **********DECK 2, CARD 1**********                       DECK2     2      P  ACTIVE
DECK2    **********DECK 2, CARD 2**********                       DECK2     3      P  ACTIVE
DECK2    *CALL COM2                                               DECK2     4      P  ACTIVE
DECK2    **********DECK 2, CARD 3**********                       DECK2     5      P  ACTIVE
DECK2    **********DECK 2, CARD 4**********                       DECK2     6      P  ACTIVE
DECK2    **********DECK 2, CARD 5**********                       DECK2     7      P  ACTIVE

DECK3    *CALL COM1                                               DECK3     4
DECK3    *CALL COM2                                               DECK3     5
```

CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

COM1 COM2 DECK1 DECK2 DECK3 DECK4 CORA

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANK$$$ COM1 COM2 DECK1 DECK3 DECK4

COMMON DECKS ENCOUNTERED

COM1 COM2

DECKS WRITTEN TO COMPILE FILE

DECK3 DECK4

THIS UPDATE REQUIRED 370000B WORDS OF CORE.

```
**********DECK 3, CARD 1**********
**********DECK 3, CARD 2**********
         COMMON DECK 1, CARD 1
         COMMON DECK 1, CARD 2
         COMMON DECK 1, CARD 3
         COMMON DECK 2, CARD 1
         COMMON DECK 2, CARD 2
         COMMON DECK 2, CARD 3
         COMMON DECK 2, CARD 4
         COMMON DECK 2, CARD 5
**********DECK 3, CARD 3**********
**********DECK 3, CARD 4**********
**********DECK 3, CARD 5**********
**********DECK 3, CARD 6**********
**********DECK 3, CARD 7**********
**********DECK 3, CARD 8**********
**********DECK 3, CARD 9**********
$$$$$$$$$$DECK 4, CARD 1$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 2$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 3$$$$$$$$$$
```

**2**

```
DECK3     2
DECK3     3
COM1      2
COM1      3
COM1      4
COM2      2
COM2      3
COM2      4
COM2      5
COM2      6
DECK3     6
DECK3     7
DECK3     8
DECK3     9
DECK3    10
DECK3    11
DECK3    12
DECK4     2
DECK4     3
DECK4     4
```

**3**

```
*COMDECK COM1
         COMMON DECK 1, CARD 1
         COMMON DECK 1, CARD 2
         COMMON DECK 1, CARD 3
*COMDECK COM2
         COMMON DECK 2, CARD 1
         COMMON DECK 2, CARD 2
         COMMON DECK 2, CARD 3
         COMMON DECK 2, CARD 4
         COMMON DECK 2, CARD 5
*DECK DECK3
**********DECK 3, CARD 1**********
**********DECK 3, CARD 2**********
*CALL COM1
*CALL COM2
**********DECK 3, CARD 3**********
**********DECK 3, CARD 4**********
**********DECK 3, CARD 5**********
**********DECK 3, CARD 6**********
**********DECK 3, CARD 7**********
**********DECK 3, CARD 8**********
**********DECK 3, CARD 9**********
*DECK DECK4
$$$$$$$$$$DECK 4, CARD 1$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 2$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 3$$$$$$$$$$
```

```
UPDAT5.
USER,DRF2755,DRF.
GET(DATAPL2)
UPDATE(F,P=DATAPL2)
COPYSBF(COMPILE)
PURGE(DATAPL2)
R  7/8/9 END-OF-RECORD
*IDENT CORB
*YANK CORA
*/  CORA = *YANKDECK DECK1
*/       = *PURDECK DECK2
*/  THIS CHANGE WILL REACTIVATE DECK1
*/  DECK2 IS PERMANENTLY DEACTIVATED
Q  6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/01. 11.44.10.*11.44.08* PAGE    1

11.44.08.UPDAT5.
11.44.08.UCCR, 64020,      0.012KCDS.
11.44.08.USER,DRF2755,.
11.44.08.GET(DATAPL2)
11.44.09.UPDATE(F,P=DATAPL2)      1
11.44.09.DECK STRUCTURE CHANGED
11.44.09. UPDATE COMPLETE.
11.44.09.COPYSBF(COMPILE)
11.44.10. EOI ENCOUNTERED.        2
11.44.10.PURGE(DATAPL2)
11.44.10.UEAD,       0.001KUNS.
11.44.10.UEPF,       0.014KUNS.
11.44.10.UFMS,       0.324KUNS.
11.44.10.UECP,       0.094SECS.
11.44.10.AESR,       1.162UNTS.
11.44.10.$OUT(*/OP=E)
11.44.10. NO FILES PROCESSED.
11.44.10.$DAYFILE(OUTPUT,JT=D)
11.44.16.UCLP, 64,023,      0.320KLNS.
```

```
UNLABELED OLDPL                IDENT    CORB                    UPDATE 1.4-528.      81/10/01. 11.44.09.      PAGE   1

*****     *IDENT CORB
*****     *YANK CORA
*****     */  CORA = *YANKDECK DECK1
*****     */       = *PURDECK DECK2
*****     */  THIS CHANGE WILL REACTIVATE DECK1
*****     */  DECK2 IS PERMANENTLY DEACTIVATED

          MODIFICATIONS / CONTROL CARDS

YANK$$$   *YANK CORA                                                  CORB          1      I
YANK$$$   *YANKDECK DECK1                                             CORA          1        D



DECK1     *DECK DECK1                                                 DECK1         1    A
DECK1     **********DECK 1, CARD 1**********                          DECK1         2    A
DECK1     **********DECK 1, CARD 2**********                          DECK1         3    A
DECK1     **********DECK 1, CARD 3**********                          DECK1         4    A
DECK1     **********DECK 1, CARD 4**********                          DECK1         5    A
DECK1     **********DECK 1, CARD 5**********                          DECK1         6    A
DECK1     **********DECK 1, CARD 6**********                          DECK1         7    A


DECK3     *CALL COM1                                                  DECK3         4
DECK3     *CALL COM2                                                  DECK3         5


          CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

          COM1      COM2      DECK1     DECK2     DECK3     DECK4     CORA      CORB


DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

          YANK$$$   COM1      COM2      DECK1     DECK3     DECK4
```

COM1 COM2

DECKS WRITTEN TO COMPILE FILE

DECK1 DECK3 DECK4

THIS UPDATE REQUIRED 370008 WORDS OF CORE.



2

```
**********DECK 1, CARD 1**********
**********DECK 1, CARD 2**********
**********DECK 1, CARD 3**********
**********DECK 1, CARD 4**********
**********DECK 1, CARD 5**********
**********DECK 1, CARD 6**********
**********DECK 3, CARD 1**********
**********DECK 3, CARD 2**********
         COMMON DECK 1, CARD 1
         COMMON DECK 1, CARD 2
         COMMON DECK 1, CARD 3
         COMMON DECK 2, CARD 1
         COMMON DECK 2, CARD 2
         COMMON DECK 2, CARD 3
         COMMON DECK 2, CARD 4
         COMMON DECK 2, CARD 5
**********DECK 3, CARD 3**********
**********DECK 3, CARD 4**********
**********DECK 3, CARD 5**********
**********DECK 3, CARD 6**********
**********DECK 3, CARD 7**********
**********DECK 3, CARD 8**********
**********DECK 3, CARD 9**********
$$$$$$$$$$DECK 4, CARD 1$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 2$$$$$$$$$$
$$$$$$$$$$DECK 4, CARD 3$$$$$$$$$$
```

```
DECK1    2
DECK1    3
DECK1    4
DECK1    5
DECK1    6
DECK1    7
DECK3    2
DECK3    3
COM1     2
COM1     3
COM1     4
COM2     2
COM2     3
COM2     4
COM2     5
COM2     6
DECK3    6
DECK3    7
DECK3    8
DECK3    9
DECK3   10
DECK3   11
DECK3   12
DECK4    2
DECK4    3
DECK4    4
```

```
UPDAT6.
USER,DRF2755,DRF.
GET(OLDPL=DATAPL1)
UPDATE(N)
SAVE(NEWPL=DATAPL3)
COPYSBF(COMPILE)
R  7/8/9 END-OF-RECORD
*IDENT COR1
*INSERT COM1.2
     COMMON DECK 1, ADDITIONAL CARD 1
     COMMON DECK 1, ADDITIONAL CARD 2
*IDENT COR2
*BEFORE DECK3.1
     //////DECK 3, ADDITIONAL CARD 1//////
     //////DECK 3, ADDITIONAL CARD 2//////
     //////DECK 3, ADDITIONAL CARD 3//////
*INSERT DECK2.4
     //// DECK 2, COMMENT-*CALL BELOW HAS BEEN INSERTED ////
*CALL COM1
     //// DECK 2, COMMENT-*CALL ABOVE HAS BEEN INSERTED ////
*IDENT COR3
*DELETE DECK1.2
Q  6/7/8/9 END-OF-INFORMATION
```

```
     FULL  DAYFILE.  81/10/01. 11.45.02.*11.45.00* PAGE    1

11.45.00.UPDAT6.
11.45.00.UCCR, 64020,      0.021KCDS.
11.45.00.USER,DRF2755,.
11.45.00.GET(OLDPL=DATAPL1)
11.45.00.UPDATE(N)                    1
11.45.02. UPDATE COMPLETE.
11.45.02.SAVE(NEWPL=DATAPL3)
11.45.02.COPYSBF(COMPILE)             2
11.45.02. EOI ENCOUNTERED.
11.45.02.UEAD,       0.001KUNS.
11.45.02.UEPF,       0.018KUNS.
11.45.02.UEMS,       0.503KUNS.
11.45.02.UECP,       0.166SECS.
11.45.02.AESR,       1.274UNTS.
11.45.02.$OUT(*/OP=E)
11.45.02. NO FILES PROCESSED.
11.45.02.$DAYFILE(OUTPUT,JT=D)
11.45.11.UCLP, 64,023,     0.448KLNS.
```

```
UNLABELED OLDPL                IDENT    COR1                    UPDATE 1.4-528.      81/10/01. 11.45.00.      PAGE   1

*****    *IDENT COR1
*****    *INSERT COM1.2
              COMMON DECK 1, ADDITIONAL CARD 1
              COMMON DECK 1, ADDITIONAL CARD 2
```

1

```
UNLABELED OLDPL                IDENT    COR2                    UPDATE 1.4-528.      81/10/01. 11.45.00.      PAGE   2

*****    *IDENT COR2
*****    *BEFORE DECK3.1
              ///////DECK 3, ADDITIONAL CARD 1///////
              ///////DECK 3, ADDITIONAL CARD 2///////
              ///////DECK 3, ADDITIONAL CARD 3///////
*****    *INSERT DECK2.4
              //// DECK 2, COMMENT-*CALL BELOW HAS BEEN INSERTED ////
*****    *CALL COM1
              //// DECK 2, COMMENT-*CALL ABOVE HAS BEEN INSERTED ////
```

```
*****    *IDENT COR3
*****    *DELETE DECK1.2


         MODIFICATIONS / CONTROL CARDS


COM1          COMMON DECK 1, ADDITIONAL CARD 1                                  COR1          1      I
COM1          COMMON DECK 1, ADDITIONAL CARD 2                                  COR1          2      I


DECK1    **********DECK 1, CARD 1**********                                     DECK1         2        D


DECK2         //// DECK 2, COMMENT-*CALL BELOW HAS BEEN INSERTED ////          COR2          1      I
DECK2    *CALL COM1                                                            COR2          2      I
DECK2         //// DECK 2, COMMENT-*CALL ABOVE HAS BEEN INSERTED ////          COR2          3      I
DECK2         //////DECK 3, ADDITIONAL CARD 1//////                           COR2          4      I
DECK2         //////DECK 3, ADDITIONAL CARD 2//////                           COR2          5      I
DECK2         //////DECK 3, ADDITIONAL CARD 3//////                           COR2          6      I


DECK2    *CALL COM2                                                            DECK2         4
DECK2    *CALL COM1                                                            COR2          2
DECK3    *CALL COM1                                                            DECK3         4
DECK3    *CALL COM2                                                            DECK3         5


              CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

         COM1      COM2      DECK1     DECK2     DECK3     DECK4     COR1      COR2
         COR3


DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

         YANK$$$   COM1      COM2      DECK1     DECK2     DECK3     DECK4
```

COM1 COM2

DECKS WRITTEN TO COMPILE FILE

DECK1 DECK2 DECK3

THIS UPDATE REQUIRED 37000B WORDS OF CORE.

2

```
**********DECK 1, CARD 2**********                                  DECK1      3
**********DECK 1, CARD 3**********                                  DECK1      4
**********DECK 1, CARD 4**********                                  DECK1      5
**********DECK 1, CARD 5**********                                  DECK1      6
**********DECK 1, CARD 6**********                                  DECK1      7
**********DECK 2, CARD 1**********                                  DECK2      2
**********DECK 2, CARD 2**********                                  DECK2      3
         COMMON DECK 2, CARD 1                                      COM2       2
         COMMON DECK 2, CARD 2                                      COM2       3
         COMMON DECK 2, CARD 3                                      COM2       4
         COMMON DECK 2, CARD 4                                      COM2       5
         COMMON DECK 2, CARD 5                                      COM2       6
     //// DECK 2, COMMENT-*CALL BELOW HAS BEEN INSERTED ////       COR2       1
         COMMON DECK 1, CARD 1                                      COM1       2
     COMMON DECK 1, ADDITIONAL CARD 1                               COR1       1
     COMMON DECK 1, ADDITIONAL CARD 2                               COR1       2
         COMMON DECK 1, CARD 2                                      COM1       3
         COMMON DECK 1, CARD 3                                      COM1       4
     //// DECK 2, COMMENT-*CALL ABOVE HAS BEEN INSERTED ////       COR2       3
**********DECK 2, CARD 3**********                                  DECK2      5
**********DECK 2, CARD 4**********                                  DECK2      6
**********DECK 2, CARD 5**********                                  DECK2      7
     //////DECK 3, ADDITIONAL CARD 1//////                         COR2       4
     //////DECK 3, ADDITIONAL CARD 2//////                         COR2       5
     //////DECK 3, ADDITIONAL CARD 3//////                         COR2       6
**********DECK 3, CARD 1**********                                  DECK3      2
**********DECK 3, CARD 2**********                                  DECK3      3
         COMMON DECK 1, CARD 1                                      COM1       2
     COMMON DECK 1, ADDITIONAL CARD 1                               COR1       1
     COMMON DECK 1, ADDITIONAL CARD 2                               COR1       2
         COMMON DECK 1, CARD 2                                      COM1       3
         COMMON DECK 1, CARD 3                                      COM1       4
         COMMON DECK 2, CARD 1                                      COM2       2
         COMMON DECK 2, CARD 2                                      COM2       3
         COMMON DECK 2, CARD 3                                      COM2       4
         COMMON DECK 2, CARD 4                                      COM2       5
         COMMON DECK 2, CARD 5                                      COM2       6
**********DECK 3, CARD 3**********                                  DECK3      6
**********DECK 3, CARD 4**********                                  DECK3      7
**********DECK 3, CARD 5**********                                  DECK3      8
**********DECK 3, CARD 6**********                                  DECK3      9
**********DECK 3, CARD 7**********                                  DECK3     10
**********DECK 3, CARD 8**********                                  DECK3     11
**********DECK 3, CARD 9**********
```

```
UPDAT7.
USER,DRF2755,DRF.
GET(OLDPL=DATAPL3)
UPDATE(N)
SAVE(NEWPL=DATAPL4)
COPYSBF(COMPILE)
PURGE(DATAPL3)
R  7/8/9 END-OF-RECORD
*IDENT COR4
*YANK COR1
*/  COR1 = *INSERT COM1.2
*/       = COMMON DECK 1, ADDITIONAL CARD 1
*/       = COMMON DECK 1, ADDITIONAL CARD 2
*IDENT COR5
*SELYANK DECK3.COR2
*/  COR2 = *BEFORE DECK3.1
*/       = ////// DECK 3, ADDITIONAL CARD 1//////
*/       = ////// DECK 3, ADDITIONAL CARD 2//////
*/       = ////// DECK 3, ADDITIONAL CARD 3//////
Q  6/7/8/9 END-OF-INFORMATION
```

```
   FULL  DAYFILE.  81/10/01. 11.45.49.*11.45.47* PAGE    1

11.45.47.UPDAT7.
11.45.47.UCCR, 64020,      0.018KCDS.
11.45.47.USER,DRF2755,.
11.45.47.GET(OLDPL=DATAPL3)
11.45.47.UPDATE(N)
11.45.48. UPDATE COMPLETE.
11.45.48.SAVE(NEWPL=DATAPL4)
11.45.49.COPYSBF(COMPILE)
11.45.49. EOI ENCOUNTERED.
11.45.49.PURGE(DATAPL3)
11.45.49.UEAD,       0.001KUNS.
11.45.49.UEPF,       0.023KUNS.
11.45.49.UEMS,       0.464KUNS.
11.45.49.UECP,       0.139SECS.
11.45.49.AESR,       1.238UNTS.
11.45.49.$OUT(*/OP=E)
11.45.49. NO FILES PROCESSED.
11.45.49.$DAYFILE(OUTPUT,JI=D)
11.45.56.UCLP, 64,023,      0.384KLNS.
```

1

2

```
*****    *IDENT COR4
*****    *YANK COR1
*****    */  COR1 = *INSERT COM1.2
*****    */       = COMMON DECK 1, ADDITIONAL CARD 1
*****    */       = COMMON DECK 1, ADDITIONAL CARD 2
```



```
*****    *IDENT COR5
*****    *SELYANK DECK3.COR2
*****    */  COR2 = *BEFORE DECK3.1
*****    */       = ////// DECK 3, ADDITIONAL CARD 1//////
*****    */       = ////// DECK 3, ADDITIONAL CARD 2//////
*****    */       = ////// DECK 3, ADDITIONAL CARD 3//////
```

MODIFICATIONS / CONTROL CARDS

```
YANK$$$  *SELYANK DECK3.COR2                                  COR5    1    I
YANK$$$  *YANK COR1                                           COR4    1    I

COM1          COMMON DECK 1, ADDITIONAL CARD 1                COR1    1    D
COM1          COMMON DECK 1, ADDITIONAL CARD 2                COR1    2    D

DECK2    *CALL COM2                                           DECK2   4
DECK2    *CALL COM1                                           COR2    2
DECK3    *CALL COM1                                           DECK3   4
DECK3    *CALL COM2                                           DECK3   5
```

CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

COM1 COM2 DECK1 DECK2 DECK3 DECK4 COR1 COR2
COR3 COR4 COR5

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANK$$$ COM1 COM2 DECK1 DECK2 DECK3 DECK4

COMMON DECKS ENCOUNTERED

COM1 COM2

DECK2 DECK3

THIS UPDATE REQUIRED 370008 WORDS OF CORE.

```
**********DECK 2, CARD 1**********                                  DECK2      2
**********DECK 2, CARD 2**********                                  DECK2      3
         COMMON DECK 2, CARD 1                                      COM2       2
         COMMON DECK 2, CARD 2                                      COM2       3
         COMMON DECK 2, CARD 3                                      COM2       4
         COMMON DECK 2, CARD 4                                      COM2       5
         COMMON DECK 2, CARD 5                                      COM2       6
     //// DECK 2, COMMENT-*CALL BELOW HAS BEEN INSERTED ////       COR2       1
         COMMON DECK 1, CARD 1                                      COM1       2
         COMMON DECK 1, CARD 2                                      COM1       3
         COMMON DECK 1, CARD 3                                      COM1       4
     //// DECK 2, COMMENT-*CALL ABOVE HAS BEEN INSERTED ////       COR2       3
**********DECK 2, CARD 3**********                                  DECK2      5
**********DECK 2, CARD 4**********                                  DECK2      6
**********DECK 2, CARD 5**********                                  DECK2      7
     ///////DECK 3, ADDITIONAL CARD 1///////                       COR2       4
     ///////DECK 3, ADDITIONAL CARD 2///////                       COR2       5
     ///////DECK 3, ADDITIONAL CARD 3///////                       COR2       6
**********DECK 3, CARD 1**********                                  DECK3      2
**********DECK 3, CARD 2**********                                  DECK3      3
         COMMON DECK 1, CARD 1                                      COM1       2
         COMMON DECK 1, CARD 2                                      COM1       3
         COMMON DECK 1, CARD 3                                      COM1       4
         COMMON DECK 2, CARD 1                                      COM2       2
         COMMON DECK 2, CARD 2                                      COM2       3
         COMMON DECK 2, CARD 3                                      COM2       4
         COMMON DECK 2, CARD 4                                      COM2       5
         COMMON DECK 2, CARD 5                                      COM2       6
**********DECK 3, CARD 3**********                                  DECK3      6
**********DECK 3, CARD 4**********                                  DECK3      7
**********DECK 3, CARD 5**********                                  DECK3      8
**********DECK 3, CARD 6**********                                  DECK3      9
**********DECK 3, CARD 7**********                                  DECK3     10
**********DECK 3, CARD 8**********                                  DECK3     11
**********DECK 3, CARD 9**********                                  DECK3     12
```

2

```
UPDATE.
USER,DRF2755,DRF.
GET(OLDPL=DATAPL4)
UPDATE(Q,S)
COPYSBF(COMPILE)
COPYSBF(SOURCE)
PURGE(DATAPL4)
R  7/8/9 END-OF-RECORD
*IDENT COR6
*YANK COR4
*/  COR4 = *YANK COR1
*/  COR1 = *INSERT COM1.2
*/       = COMMON DECK 1, ADDITIONAL CARD 1
*/       = COMMON DECK 1, ADDITIONAL CARD 2
*RESTORE DECK1.2
*COMPILE DECK1,DECK2
Q  6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/01. 11.46.27.*11.46.26* PAGE    1

11.46.26.UPDAT8.
11.46.26.UCCR, 64020,       0.015KCDS.
11.46.26.USER,DRF2755,.
11.46.26.GET(OLDPL=DATAPL4)
11.46.26.UPDATE(Q,S)                    1
11.46.27. UPDATE COMPLETE.
11.46.27.COPYSBF(COMPILE)               2
11.46.27. EOI ENCOUNTERED.
11.46.27.COPYSBF(SOURCE)                  3
11.46.27. EOI ENCOUNTERED.
11.46.27.PURGE(DATAPL4)
11.46.27.UEAD,        0.001KUNS.
11.46.27.UEPF,        0.014KUNS.
11.46.27.UEMS,        0.391KUNS.
11.46.27.UECP,        0.126SECS.
11.46.27.AESR,        1.206UNTS.
11.46.27.$OUT(*/OP=E)
11.46.27. NO FILES PROCESSED.
11.46.27.$DAYFILE(OUTPUT,JT=D)
11.46.35.UCLP, 64,023,      0.384KLNS.
```

```
*****    *IDENT COR6
*****    *YANK COR4
*****    */  COR4 = *YANK COR1
*****    */  COR1 = *INSERT COM1.2
*****    */       = COMMON DECK 1, ADDITIONAL CARD 1
*****    */       = COMMON DECK 1, ADDITIONAL CARD 2
*****    *RESTORE DECK1.2
*****    *COMPILE DECK1,DECK2


         MODIFICATIONS / CONTROL CARDS

YANK$$$  *YANK COR4                                          COR6          1      I
YANK$$$  *SELYANK DECK3.COR2                                 COR5          1
YANK$$$  *YANK COR1                                          COR4          1        D


COM1          COMMON DECK 1, ADDITIONAL CARD 1               COR1          1    A
COM1          COMMON DECK 1, ADDITIONAL CARD 2               COR1          2    A


DECK1    **********DECK 1, CARD 1**********                  DECK1         2    A


DECK2    *CALL COM2                                         DECK2         4
DECK2    *CALL COM1                                         COR2          2


         CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

         COM1      COM2      DECK1     DECK2     DECK3     DECK4     COR1      COR2
         COR3      COR4      COR5      COR6


          DECK LIST AS READ FROM OLDPL PLUS ADDED NEW DECKS

         YANK$$$   COM1      COM2      DECK1     DECK2     DECK3     DECK4
```

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANK$$$ COM1 COM2 DECK1 DECK2

COMMON DECKS ENCOUNTERED

COM1 COM2

DECKS WRITTEN TO COMPILE FILE

DECK1 DECK2

THIS UPDATE REQUIRED 37000B WORDS OF CORE.



```
**********DECK 1, CARD 1**********                    DECK1   2
**********DECK 1, CARD 2**********                    DECK1   3
**********DECK 1, CARD 3**********                    DECK1   4
**********DECK 1, CARD 4**********                    DECK1   5
**********DECK 1, CARD 5**********                    DECK1   6
**********DECK 1, CARD 6**********                    DECK1   7
**********DECK 2, CARD 1**********                    DECK2   2
**********DECK 2, CARD 2**********                    DECK2   3
         COMMON DECK 2, CARD 1                        COM2    2
         COMMON DECK 2, CARD 2                        COM2    3
         COMMON DECK 2, CARD 3                        COM2    4
         COMMON DECK 2, CARD 4                        COM2    5
         COMMON DECK 2, CARD 5                        COM2    6
     //// DECK 2, COMMENT-*CALL BELOW HAS BEEN INSERTED ////  COR2    1
         COMMON DECK 1, CARD 1                        COM1    2
     COMMON DECK 1, ADDITIONAL CARD 1                 COR1    1
     COMMON DECK 1, ADDITIONAL CARD 2                 COR1    2
         COMMON DECK 1, CARD 2                        COM1    3
         COMMON DECK 1, CARD 3                        COM1    4
     //// DECK 2, COMMENT-*CALL ABOVE HAS BEEN INSERTED ////  COR2    3
**********DECK 2, CARD 3**********                    DECK2   5
**********DECK 2, CARD 4**********                    DECK2   6
**********DECK 2, CARD 5**********                    DECK2   7
     //////DECK 3, ADDITIONAL CARD 1//////            COR2    4
     //////DECK 3, ADDITIONAL CARD 2//////            COR2    5
     //////DECK 3, ADDITIONAL CARD 3//////            COR2    6
```

2

```
*COMDECK COM1
          COMMON DECK 1, CARD 1
     COMMON DECK 1, ADDITIONAL CARD 1
     COMMON DECK 1, ADDITIONAL CARD 2
          COMMON DECK 1, CARD 2
          COMMON DECK 1, CARD 3
*COMDECK COM2
          COMMON DECK 2, CARD 1
          COMMON DECK 2, CARD 2
          COMMON DECK 2, CARD 3
          COMMON DECK 2, CARD 4
          COMMON DECK 2, CARD 5
*DECK DECK1
**********DECK 1, CARD 1**********
**********DECK 1, CARD 2**********
**********DECK 1, CARD 3**********
**********DECK 1, CARD 4**********
**********DECK 1, CARD 5**********
**********DECK 1, CARD 6**********
*DECK DECK2
**********DECK 2, CARD 1**********
**********DECK 2, CARD 2**********
*CALL COM2
     //// DECK 2, COMMENT-*CALL BELOW HAS BEEN INSERTED ////
*CALL COM1
     //// DECK 2, COMMENT-*CALL ABOVE HAS BEEN INSERTED ////
**********DECK 2, CARD 3**********
**********DECK 2, CARD 4**********
**********DECK 2, CARD 5**********
     //////DECK 3, ADDITIONAL CARD 1//////
     //////DECK 3, ADDITIONAL CARD 2//////
     //////DECK 3, ADDITIONAL CARD 3//////
```

3

```
UPDAT9.
USER,DRF2755,DRF.
GET(OLDPL=DATAPL1)
UPDATE(N)
SAVE(NEWPL=DATAPL5)
COPYSBF(COMPILE)
PURGE(DATAPL1)
R  7/8/9 END-OF-RECORD
*IDENT COR7
*INSERT COM2.2
          COMMON DECK 2, ADDITIONAL CARD 1
*IDENT CORA
*DELETE DECK1.2,DECK1.4
     /////DECK 1, REPLACEMENT CARD 1/////
     /////DECK 1, REPLACEMENT CARD 2/////
     /////DECK 1, REPLACEMENT CARD 3/////
Q  6/7/8/9 END-OF-INFORMATION
```

```
     FULL  DAYFILE.  81/10/01. 11.46.41.*11.46.39* PAGE    1

11.46.39.UPDAT9.
11.46.39.UCCR, 64020,      0.015KCDS.
11.46.39.USER,DRF2755,.
11.46.39.GET(OLDPL=DATAPL1)
11.46.39.UPDATE(N)                    1
11.46.41. UPDATE COMPLETE.
11.46.41.SAVE(NEWPL=DATAPL5)
11.46.41.COPYSBF(COMPILE)             2
11.46.41. EOI ENCOUNTERED.
11.46.41.PURGE(DATAPL1)
11.46.41.UEAD,       0.001KUNS.
11.46.41.UEPF,       0.023KUNS.
11.46.41.UEMS,       0.456KUNS.
11.46.41.UECP,       0.144SECS.
11.46.41.AESR,       1.240UNTS.
11.46.41.$OUT(*/OP=E)
11.46.41. NO FILES PROCESSED.
11.46.41.$DAYFILE(OUTPUT,JT=D)
11.46.48.UCLP, 64,023,      0.384KLNS.
```

```
UNLABELED OLDPL              IDENT    COR7                    UPDATE 1.4-528.      81/10/01. 11.46.39.      PAGE   1

*****    *IDENT COR7
*****    *INSERT COM2.2
                COMMON DECK 2, ADDITIONAL CARD 1
```

1

```
UNLABELED OLDPL              IDENT    COR8                    UPDATE 1.4-528.      81/10/01. 11.46.39.      PAGE   2

*****    *IDENT COR8
*****    *DELETE DECK1.2,DECK1.4
             //////DECK 1, REPLACEMENT CARD 1//////
             //////DECK 1, REPLACEMENT CARD 2//////
             //////DECK 1, REPLACEMENT CARD 3//////


         MODIFICATIONS / CONTROL CARDS


COM2             COMMON DECK 2, ADDITIONAL CARD 1                    COR7          1      I


DECK1    **********DECK 1, CARD 1**********                          DECK1         2        D
DECK1    **********DECK 1, CARD 2**********                          DECK1         3        D
DECK1    **********DECK 1, CARD 3**********                          DECK1         4        D
DECK1        //////DECK 1, REPLACEMENT CARD 1//////                  COR8          1      I
DECK1        //////DECK 1, REPLACEMENT CARD 2//////                  COR8          2      I
DECK1        //////DECK 1, REPLACEMENT CARD 3//////                  COR8          3      I


DECK2    *CALL COM2                                                  DECK2         4
DECK3    *CALL COM1                                                  DECK3         4
DECK3    *CALL COM2                                                  DECK3         5


              CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

         COM1      COM2      DECK1     DECK2     DECK3     DECK4     COR7      COR8


DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

         YANK$$$   COM1      COM2      DECK1     DECK2     DECK3     DECK4
```

COM1 COM2

DECKS WRITTEN TO COMPILE FILE

DECK1 DECK2 DECK3

THIS UPDATE REQUIRED 370008 WORDS OF CORE.

```
     //////DECK 1, REPLACEMENT CARD 1//////          COR8      1
     //////DECK 1, REPLACEMENT CARD 2//////          COR8      2
     //////DECK 1, REPLACEMENT CARD 3//////          COR8      3
**********DECK 1, CARD 4**********                   DECK1     5
**********DECK 1, CARD 5**********                   DECK1     6
**********DECK 1, CARD 6**********                   DECK1     7
**********DECK 2, CARD 1**********                   DECK2     2
**********DECK 2, CARD 2**********                   DECK2     3
          COMMON DECK 2, CARD 1                     COM2      2
           COMMON DECK 2, ADDITIONAL CARD 1         COR7      1
          COMMON DECK 2, CARD 2                     COM2      3
          COMMON DECK 2, CARD 3                     COM2      4
          COMMON DECK 2, CARD 4                     COM2      5
          COMMON DECK 2, CARD 5                     COM2      6
**********DECK 2, CARD 3**********                   DECK2     5
**********DECK 2, CARD 4**********                   DECK2     6
**********DECK 2, CARD 5**********                   DECK2     7
**********DECK 3, CARD 1**********                   DECK3     2
**********DECK 3, CARD 2**********                   DECK3     3
          COMMON DECK 1, CARD 1                     COM1      2
          COMMON DECK 1, CARD 2                     COM1      3
          COMMON DECK 1, CARD 3                     COM1      4
          COMMON DECK 2, CARD 1                     COM2      2
           COMMON DECK 2, ADDITIONAL CARD 1         COR7      1
          COMMON DECK 2, CARD 2                     COM2      3
          COMMON DECK 2, CARD 3                     COM2      4
          COMMON DECK 2, CARD 4                     COM2      5
          COMMON DECK 2, CARD 5                     COM2      6
**********DECK 3, CARD 3**********                   DECK3     6
**********DECK 3, CARD 4**********                   DECK3     7
**********DECK 3, CARD 5**********                   DECK3     8
**********DECK 3, CARD 6**********                   DECK3     9
**********DECK 3, CARD 7**********                   DECK3    10
**********DECK 3, CARD 8**********                   DECK3    11
**********DECK 3, CARD 9**********                   DECK3    12
```

2

```
UPDATA.
USER,DRF2755,DRF.
GET(OLDPL=DATAPL5)
UPDATE(F)
COPYSBF(COMPILE)
PURGE(DATAPL5)
CATLIST.
R  7/8/9 END-OF-RECORD
*PURGE COR7,COR8
*/  COR7 = *INSERT COM2.?
*/       = COMMON DECK 2, ADDITIONAL CARD 1
*/  COR8 = *DELETE DECK1.2,DECK1.4
*/       = /////DECK1, REPLACEMENT CARD 1/////
*/       = /////DECK1, REPLACEMENT CARD 2/////
*/       = /////DECK1, REPLACEMENT CARD 3/////
Q  6/7/8/9 END-OF-INFORMATION
```



```
     FULL  DAYFILE.  81/10/01. 11.46.52.*11.46.50* PAGE    1

11.46.50.UPDATA.
11.46.50.UCCR, 64020,       0.014KCDS.
11.46.50.USER,DRF2755,.
11.46.50.GET(OLDPL=DATAPL5)
11.46.51.UPDATE(F)
11.46.52. UPDATE COMPLETE.
11.46.52.COPYSBF(COMPILE)
11.46.52. EOI ENCOUNTERED.
11.46.52.PURGE(DATAPL5)
11.46.52.CATLIST.
11.46.52. CATLIST COMPLETE.
11.46.52.UEAD,       0.001KUNS.
11.46.52.UEPF,       0.019KUNS.
11.46.52.UEMS,       0.367KUNS.
11.46.52.UECP,       0.116SECS.
11.46.52.AESR,       1.193UNTS.
11.46.52.$OUT(*/OP=E)
11.46.52. NO FILES PROCESSED.
11.46.52.$DAYFILE(OUTPUT,JT=D)
11.47.01.UCLP, 64,023,      0.384KLNS.
```

1 2 3

```
*****    *PURGE COR7,COR8
*****    */  COR7 = *INSERT COM2.2
*****    */       = COMMON DECK 2, ADDITIONAL CARD 1
*****    */  COR8 = *DELETE DECK1.2,DECK1.4
*****    */       = /////DECK1, REPLACEMENT CARD 1/////
*****    */       = /////DECK1, REPLACEMENT CARD 2/////
*****    */       = /////DECK1, REPLACEMENT CARD 3/////
```

1

MODIFICATIONS / CONTROL CARDS

```
COM2              COMMON DECK 2, ADDITIONAL CARD 1                    COR7          1          P  ACTIVE


DECK1    **********DECK 1, CARD 1**********                           DECK1         2    A
DECK1    **********DECK 1, CARD 2**********                           DECK1         3    A
DECK1    **********DECK 1, CARD 3**********                           DECK1         4    A
DECK1        /////DECK 1, REPLACEMENT CARD 1/////                     COR8          1          P  ACTIVE
DECK1        /////DECK 1, REPLACEMENT CARD 2/////                     COR8          2          P  ACTIVE
DECK1        /////DECK 1, REPLACEMENT CARD 3/////                     COR8          3          P  ACTIVE


DECK2    *CALL COM2                                                  DECK2         4


DECK3    *CALL COM1                                                  DECK3         4
DECK3    *CALL COM2                                                  DECK3         5
```

CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

COM1 COM2 DECK1 DECK2 DECK3 DECK4

2 PURGED IDENTS WERE FOUND

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANK$$$ COM1 COM2 DECK1 DECK2 DECK3 DECK4

COMMON DECKS ENCOUNTERED

COM1 COM2

DECKS WRITTEN TO COMPILE FILE

DECK1 DECK2 DECK3 DECK4

THIS UPDATE REQUIRED 370008 WORDS OF CORE.

```
**********DECK 1, CARD 1**********                    DECK1       2
**********DECK 1, CARD 2**********                    DECK1       3
**********DECK 1, CARD 3**********                    DECK1       4
**********DECK 1, CARD 4**********                    DECK1       5
**********DECK 1, CARD 5**********                    DECK1       6
**********DECK 1, CARD 6**********                    DECK1       7
**********DECK 2, CARD 1**********                    DECK2       2
**********DECK 2, CARD 2**********                    DECK2       3
         COMMON DECK 2, CARD 1                       COM2        2
         COMMON DECK 2, CARD 2                       COM2        3
         COMMON DECK 2, CARD 3                       COM2        4
         COMMON DECK 2, CARD 4                       COM2        5
         COMMON DECK 2, CARD 5                       COM2        6
**********DECK 2, CARD 3**********                    DECK2       5
**********DECK 2, CARD 4**********                    DECK2       6
**********DECK 2, CARD 5**********                    DECK2       7
**********DECK 3, CARD 1**********                    DECK3       2
**********DECK 3, CARD 2**********                    DECK3       3
         COMMON DECK 1, CARD 1                       COM1        2
         COMMON DECK 1, CARD 2                       COM1        3
         COMMON DECK 1, CARD 3                       COM1        4
         COMMON DECK 2, CARD 1                       COM2        2
         COMMON DECK 2, CARD 2                       COM2        3
         COMMON DECK 2, CARD 3                       COM2        4
         COMMON DECK 2, CARD 4                       COM2        5
         COMMON DECK 2, CARD 5                       COM2        6
**********DECK 3, CARD 3**********                    DECK3       6
**********DECK 3, CARD 4**********                    DECK3       7
**********DECK 3, CARD 5**********                    DECK3       8
**********DECK 3, CARD 6**********                    DECK3       9
**********DECK 3, CARD 7**********                    DECK3      10
**********DECK 3, CARD 8**********                    DECK3      11
**********DECK 3, CARD 9**********                    DECK3      12
$$$$$$$$$$DECK 4, CARD 1$$$$$$$$$$                    DECK4       2
$$$$$$$$$$DECK 4, CARD 2$$$$$$$$$$                    DECK4       3
$$$$$$$$$$DECK 4, CARD 3$$$$$$$$$$                    DECK4       4
```

2



```
CATALOG OF  DRF2755          FM/FEATURE 81/10/01. 11.46.52.  PAGE   1

INDIRECT ACCESS FILE(S)

 DEMO      .SEG       UPDATEC    UPDATED    USELIB1    USELIB2    USELIB3
 LOAD1      TESTCOP

DIRECT ACCESS FILE(S)

 CKPFL

        9 INDIRECT ACCESS FILE(S),    TOTAL PRUS =        239.
        1 DIRECT ACCESS FILE(S),      TOTAL PRUS =          0.
```

3

# EXERCISE 10

PROBLEM

GIVEN:

1. A FORTRAN or COBOL program deck with some statements in error and some missing statements.

2. Data that is input to the program.

3. The missing statements and corrections for the program.

Below is a program you can use if one is not provided or you may write your own.

PROBLEM: Set up the deck to run in Batch Mode to do the following:

1. Place the program on a permanent file in source library format and compile the program.

2. Replace the error cards with the corrected ones and insert the missing cards. Compile the program to be sure you have made the changes correctly. Using the supplied data cards as input to the corrected program, compile and execute it.

3. Delete the permanent file from the system and make sure the delete actually took place.

```
      PROGRAM AVE(INPUT,OUTPUT)
      DIMENSHUN X(10)
      SUM=0.
      DO 10 I=1,10                                    1.  PROGRAM IN ERROR
   11 SUM=SUM+X(I)
      AVG=SUN/10.
      PRINT 10,AVG
    1 FORMAT (1H1,*THE ANSWER IS*,F10.3)
      STOP
      END
```

```
1. 2. 3.
4. 5.                                                 2.  DATA
  7.,8.,9.,10.
```

```
      DIMENSION X(10)
      READ *,X
   10 SUM=SUM+X(I)                                    3.  CORRECTIONS
      AVG=SUM/10.
      PRINT 1,AVG
```

LESSON GUIDE 6

# INTRODUCTION TO LIBRARY GENERATION

## LESSON PREVIEW:

This lesson is intended to provide an introduction to binary library creation.

## REFERENCES:

NOS Version 1 Reference Manual, Volume 1, Section 7 (LIBGEN statement).

## TRAINING AIDS:

Visuals V6-1 through V6-14.

## PROJECTS:

Exercise 11 (problem).

## OBJECTIVES:

Upon the completion of this lesson, the student will:

1. Have a basic understanding of binary library creation.
2. Be able to set up a simple job, to create a library.

| | |
|---|---|
| Pi | any of the following in any order: |
| F= Lfn1 | name of file containing records to be placed on user library file 1Fn2. Default file name is LGO. |
| P= Lfn2 | name of the file on which the user library is to be written. Default file name is ULIB. |
| N= Lfn3 | name of the user library being generated, this name becomes the name of the ULIB and OPDL records. Default is Lfn3=Lfn2. |
| NX=n | if n is nonzero, no cross references are given. Decks are not cross-linked in the ULIB directory (avoid duplicate entry points on loads). Default is NX=0. |

THE APPLICABLE MEANING OF THE TERM LIBRARY MUST BE DETERMINED FROM ITS CONTEXT.

### USER NAME LIBRARY

FILES STORED UNDER USER NAME LIBRARY NEED NOT BE LIBRARIES THEMSELVES.

### PROGRAM LIBRARIES

A PROGRAM LIBRARY IS A COLLECTION OF COMPRESSED SOURCE DECK IMAGES STORED IN MODIFY OR UPDATE FORMAT.

### USER LIBRARIES

USER LIBRARIES ARE THE FILES NAMED IN THE SATISFY, LIBLOAD, OR LOSET LOADER COMMAND OR IN PROGRAM BINARIES. THESE FILES ARE SEARCHED BY CYBER LOADER TO SATISFY EXTERNAL REFERENCES WITHIN A PROGRAM IT IS LOADING.

### GLOBAL LIBRARIES

GLOBAL LIBRARIES ARE THE FILES NAMED ON THE LIBRARY COMMAND. THE FILES ARE SEARCHED:

1) FOR A PROGRAM NAME, OR PROCEDURE NAME THAT MATCHES THE COMMAND NAME.

2) BY CYBER LOADER TO SATISFY EXTERNAL REFERENCES WITHIN A PROGRAM IT IS LOADING.

### CENTRAL LIBRARY

ROUTINES FROM THE SYSTEM TAPE.

IN THIS LESSON, LIBRARY WILL REFER TO USER OR GLOBAL LIBRARIES ONLY.

A LIBRARY IS A FILE CONTAINING RECORDS THAT ARE ACCESSED INDIVIDUALLY. LIBRARY RECORDS CAN BE OF SEVERAL TYPES, INCLUDING:

| | |
|---|---|
| ABS | MULTIPLE ENTRY POINT OVERLAY |
| CAP | CYBER LOADER CAPSULE |
| PROC | PROCEDURE FILE |
| REL | RELOCATABLE USER PROGRAM |
| ULIB | USER LIBRARY DIRECTORY |

FOR THE SEARCH ORDER OF THE CYBER LOADER SEE LOADER LESSON.

WHEN A COMMAND IS ENCOUNTERED BY THE SYSTEM, THE SEARCH ORDER IS:

1) IF COMMAND PREFIXED BY A $, GO TO STEP 4.

2) SEARCH THE USER'S LOCAL FILES FOR A MATCHING FILE NAME.

3) SEARCH THE GLOBAL LIBRARY DIRECTORY FOR A PROGRAM NAME THAT MATCHES THE COMMAND NAME.

4) SEARCH THE CENTRAL LIBRARY DIRECTORY FOR A MATCHING PROGRAM NAME.

THE FORMAT OF THE LIBRARY COMMAND IS:

LIBRARY, LFN 1, LFN 2.

SET THE GLOBAL LIBRARY TO THE CONTENTS OF FILES LFN 1, AND LFN 2.

LIBRARY, LFN 3/A.

APPENDS THE CONTENTS OF FILE LFN 3 TO THE GLOBAL LIBRARY.

LIBRARY.

CLEARS THE GLOBAL LIBRARY.

EXAMPLE 1:

USING LIBEDIT TO CREATE A LIBRARY.

LIBEDIT DOESN'T CREATE LIBRARIES. THE U PARAMETER CALLS LIBGEN WHICH CREATES THE LIBRARY.

```
LIBEDIT,P=0,N=TESTLI,B=PROCFIL,U

ENTER DIRECTIVES -

? *BUILD,LIBR

? *B,*,PROC/*

?

      RECORDS WRITTEN ON FILE TESTLI

        RECORD      TYPE      FILE      DATE      COMMENT

INSERTED CREATE     PROC      PROCFIL
INSERTED CRESAVE    PROC      PROCFIL
INSERTED CTFILE     PROC      PROCFIL
INSERTED CZFILE     PROC      PROCFIL
INSERTED CWFILE     PROC      PROCFIL
INSERTED IWFILE     PROC      PROCFIL
INSERTED FDUMP      PROC      PROCFIL
INSERTED DM         PROC      PROCFIL
INSERTED SORTMRG    PROC      PROCFIL
ADDED    FILE,CT    TEXT      PROCFIL
ADDED    FILE,CW    TEXT      PROCFIL
ADDED    LIBR       OPLD      *****     83/03/17.

LIBRARY GENERATION COMPLETE.

/SAVE,TESTLI
/LIBRARY,TESTLI
LIBRARY, TESTLI.
/CRESAVE

DMF ALREADY PERMANENT.
```

```
LOAD1,T5.
USER,DRF2755,DRF.
MAP(PART)
FTN,R=0,GO.
SAVE(LGO=LOAD1)
COBOL(B=LG01)
LG01.
FTN(B=LG02,R=0,GO)
R     7/8/9 END-OF-RECORD
      PROGRAM MAIN(OUTPUT)
      PRINT 1
      CALL SUB1
      CALL SUB2
      CALL SUB3
      CALL SUB4
    1 FORMAT(1H1,*IN MAIN*)
      END
      SUBROUTINE SUB1
      PRINT *," IN SUB1-PROGRAM MAIN"
      END
      SUBROUTINE SUB2
      PRINT *," IN SUB2-PROGRAM MAIN"
      END
      SUBROUTINE SUB3
      PRINT *," IN SUB3-PROGRAM MAIN"
      END
      SUBROUTINE SUB4
      PRINT *," IN SUB4-PROGRAM MAIN"
      END
R     7/8/9 END-OF-RECORD
       IDENTIFICATION DIVISION.
       PROGRAM-ID. PARAMETER.
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
       SOURCE-COMPUTER. 6600.
       OBJECT-COMPUTER. 6600.
       SPECIAL-NAMES.
           OUTPUT-C IS PRINT.
       DATA DIVISION.
       WORKING-STORAGE SECTION.
       77  PARM1 USAGE IS COMPUTATIONAL-1 VALUE IS 3 SIZE IS 1 DIGITS.
       77  PARM2 USAGE IS COMPUTATIONAL-1 VALUE IS 4 SIZE IS 1 DIGITS.
       77  PARM3 USAGE IS COMPUTATIONAL-1 VALUE IS 0 SIZE IS 1 DIGITS.
       77  PARM4 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
       77  PARM5 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
       77  PARM6 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
       PROCEDURE DIVISION.
       BEGIN.
           DISPLAY "1PARAMETER 1 IS    " PARM1 UPON PRINT.
           DISPLAY " PARAMETER 2 IS    " PARM2 UPON PRINT.
           DISPLAY " PARAMETER 3 IS    " PARM3 UPON PRINT.
           MOVE PARM2 TO PARM3.
           ADD PARM1 TO PARM3.
           MOVE PARM1 TO PARM4.
           MOVE PARM2 TO PARM5.
           MOVE PARM3 TO PARM6.
           DISPLAY " PARAMETER 1 IS    " PARM4 UPON PRINT.
           DISPLAY " PARAMETER 2 IS    " PARM5 UPON PRINT.
           DISPLAY " THE SUM IS        " PARM6 UPON PRINT.
       STOP RUN.
```

```
      PROGRAM DON(OUTPUT)
      CALL PRG1
      CALL PRG2
C     NOTE THIS IS A DUMMY PROGRAM
C     THIS DUMMY PROGRAM IS REQUIRED TO SET
C     UP THE OUPUT BUFFER FOR THE FORTRAN
C     OBJECT TIME ROUTINES.
      END
      SUBROUTINE PRG2
      COMMON /ANSWERS/ NUMBERS(10)
      PRINT  *,(NUMBERS(I),I=1,10)
      END
          IDENT PRG1
          ENTRY PRG1
          LIST   -B,-R
OPERAND   DATA 1,2,3,4,5,6,7,8,9,10
PRG1      DATA   0
          SB1    9
LOOP      SA1    OPERAND+B1
          BX6    X1
          SA6    SUM+B1
          SB1    B1-1
          PL     B1,LOOP
          EQ     PRG1
          USE    /ANSWERS/
SUM       BSS    10
          END
0    6/7/8/9 END-OF-INFORMATION
   FULL  DAYFILE.  81/10/07. 11.43.24.*11.43.07* PAGE     1

11.43.07.LOAD1,T5.
11.43.07.UCCR, 64020,       0.085KCDS.
11.43.07.USER,DRF2755,.
11.43.07.MAP(PART)
11.43.07.FTN,R=0,GO.
11.43.12.       .125 CP SECONDS COMPILATION TIME
11.43.12.LGO
11.43.14.      END   MAIN
11.43.14.     014300 MAXIMUM EXECUTION FL.
11.43.14.      0.036 CP SECONDS EXECUTION TIME.
11.43.14.SAVE(LGO=LOAD1)
11.43.14.COBOL(B=LG01)
11.43.15.COMPILING PARAMET
11.43.17. 000 E AND      T/U  DIAGNOSTICS ISSUED
11.43.17.     0620008 SCM USED
11.43.17.       .255 CP SECONDS COMPILATION TIME
11.43.17.END COBOL
11.43.17.LG01.
11.43.19.FTN(B=LG02,R=0,GO)
11.43.22.       .264 CP SECONDS COMPILATION TIME
11.43.22.LG02
11.43.24.      END   DON
11.43.24.     013100 MAXIMUM EXECUTION FL.
11.43.24.      0.035 CP SECONDS EXECUTION TIME.
11.43.24.UEAD,        0.001KUNS.
11.43.24.UEPF,        0.005KUNS.
11.43.24.UEMS,        7.566KUNS.
11.43.24.UECP,        1.957SECS.
11.43.24.AESR,        4.820UNTS.
11.43.24.$OUT(*/OP=E)
11.43.24. NO FILES PROCESSED.
```

1
2
3
4

```
PROGRAM MAIN       73/74   OPT=1                    FTN 4.8+552       81/10/07. 11.43.07       PAGE     1

1                PROGRAM MAIN(OUTPUT)
                 PRINT 1
                 CALL SUB1
                 CALL SUB2
5                CALL SUB3
                 CALL SUB4
               1 FORMAT(1H1,*IN MAIN*)
                 END
```

1

```
SUBROUTINE SUB1       73/74   OPT=1                    FTN 4.8+552       81/10/07. 11.43.07       PAGE     1

1                SUBROUTINE SUB1
                 PRINT *,# IN SUB1-PROGRAM MAIN#
                 END
```

```
SUBROUTINE SUB2       73/74   OPT=1                    FTN 4.8+552       81/10/07. 11.43.07       PAGE     1

1                SUBROUTINE SUB2
                 PRINT *,# IN SUB2-PROGRAM MAIN#
                 END
```

```
SUBROUTINE SUB3       73/74   OPT=1                    FTN 4.8+552       81/10/07. 11.43.07       PAGE     1

1                SUBROUTINE SUB3
                 PRINT *,# IN SUB3-PROGRAM MAIN#
                 END
```

```
SUBROUTINE SUB4       73/74   OPT=1                    FTN 4.8+552       81/10/07. 11.43.07       PAGE     1

1                SUBROUTINE SUB4
                 PRINT *,# IN SUB4-PROGRAM MAIN#
                 END
```

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 10162

TRANSFER ADDRESS -- MAIN 2174

PROGRAM ENTRY POINTS -- MAIN 2174

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| MAIN | 111 | 2101 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | PROGRAM OPT=1 |
| SUB1 | 2212 | 14 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB2 | 2226 | 14 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB3 | 2242 | 14 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB4 | 2256 | 14 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SYSAID= | 2272 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2273 | 1 | | | | | | | |
| /FCL.C./ | 2274 | 30 | | | | | | | |
| /Q8.IO./ | 2324 | 130 | | | | | | | |
| Q2NTRY= | 2454 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2454 | 43 | | | | | | | |
| COMIO= | 2517 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2552 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL CAPSULE LOADING |
| FECMSK= | 2612 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2653 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2656 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON FLOATING OUTPUT CODE |
| FMTAP= | 3167 | 357 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | 3546 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 4247 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL MISC. UTILITIES. |
| GETFIT= | 4274 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | 4370 | 470 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | OUTPUT FORMAT INTERPRETER. |
| LDOUT= | 5060 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 5324 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON OUTPUT CODE |
| OUTC= | 5500 | 162 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORMATTED WRITE FORTRAN RECORD. |
| OUTF= | 5662 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 6066 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 6073 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 6133 | 14 | | | | | | | |
| FDL.RES | 6147 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 6360 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 6602 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | 7237 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| ERR$RM | 7316 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 7343 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 7432 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 7437 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 7621 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 7627 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 7643 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 7701 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 7723 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | 7732 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 10140 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

.421 CP SECONDS 260008 C[illegible] STORAGE USED 8 TABLE MOVES

```
IN MAIN
 IN SUB1-PROGRAM MAIN
 IN SUB2-PROGRAM MAIN
 IN SUB3-PROGRAM MAIN
 IN SUB4-PROGRAM MAIN
```

```
00001            IDENTIFICATION DIVISION.
00002            PROGRAM-ID. PARAMETER.
00003            ENVIRONMENT DIVISION.
00004            CONFIGURATION SECTION.
00005            SOURCE-COMPUTER. 6600.
00006            OBJECT-COMPUTER. 6600.
00007            SPECIAL-NAMES.
00008                OUTPUT-C IS PRINT.
00009            DATA DIVISION.
00010            WORKING-STORAGE SECTION.
00011            77  PARM1 USAGE IS COMPUTATIONAL-1 VALUE IS 3 SIZE IS 1 DIGITS.
00012            77  PARM2 USAGE IS COMPUTATIONAL-1 VALUE IS 4 SIZE IS 1 DIGITS.
00013            77  PARM3 USAGE IS COMPUTATIONAL-1 VALUE IS 0 SIZE IS 1 DIGITS.
00014            77  PARM4 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
00015            77  PARM5 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
00016            77  PARM6 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
```

**2**



```
00017            PROCEDURE DIVISION.
00018            BEGIN.
00019                DISPLAY #1PARAMETER 1 IS    # PARM1 UPON PRINT.
00020                DISPLAY # PARAMETER 2 IS    # PARM2 UPON PRINT.
00021                DISPLAY # PARAMETER 3 IS    # PARM3 UPON PRINT.
00022                MOVE PARM2 TO PARM3.
00023                ADD PARM1 TO PARM3.
00024                MOVE PARM1 TO PARM4.
00025                MOVE PARM2 TO PARM5.
00026                MOVE PARM3 TO PARM6.
00027                DISPLAY # PARAMETER 1 IS    # PARM4 UPON PRINT.
00028                DISPLAY # PARAMETER 2 IS    # PARM5 UPON PRINT.
00029                DISPLAY # THE SUM IS        # PARM6 UPON PRINT.
00030            STOP RUN.
PARAMET LENGTH IS  000122
 062000B SCM USED
```

3

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 5175

TRANSFER ADDRESS -- CENT.00 274

PROGRAM ENTRY POINTS -- OUTPUT= 274

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| OUTPUT= | 111 | 116 | LGO1 | 81/10/07 | COBOL | 4.7 | 552 | I | |
| /D.COMON/ | 227 | 24 | | | | | | | |
| D.COMON | 253 | 0 | LGO1 | 81/10/07 | COBOL | 4.7 | 552 | I | |
| /PARAME=/ | 253 | 21 | | | | | | | |
| PARAMET | 274 | 122 | LGO1 | 81/10/07 | COBOL | 4.7 | 552 | I | |
| C=COBIO | 416 | 751 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=CRM15 | 1367 | 10 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=DSPLY | 1377 | 427 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| /STP.END/ | 2026 | 1 | | | | | | | |
| C=NSTAN | 2027 | 155 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=TRUBL | 2204 | 121 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=CVBD | 2325 | 136 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=USE | 2463 | 211 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=ZNTAB | 2674 | 12 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=TENDP | 2706 | 73 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=TENS | 3001 | 25 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=BN | 3026 | 21 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| D=MOVE | 3047 | 133 | SL-COBOL | 81/08/10 | COMPASS | 3.6 | 552 | | |
| CPU.SYS | 3202 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| CTL$RM | 3242 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CPU.CPM | 3677 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| /FDL.COM/ | 3704 | 14 | | | | | | | |
| FDL.MMI | 3720 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| ERR$RM | 4142 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 4167 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 4256 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 4263 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.FRF | 4445 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| FDL.RES | 4503 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| CMF.CSF | 4714 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 4722 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMM.R | 4736 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 5144 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |
| CMM.MEM | 5166 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |

.409 CP SECONDS  223008 CM STORAGE USED  3 TABLE MOVES

PARAMETER 1 IS 3
PARAMETER 2 IS 4
PARAMETER 3 IS 0
PARAMETER 1 IS 3
PARAMETER 2 IS 4
THE SUM IS 7

```
    PROGRAM DON        73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.43.19        PAGE     1

1                   PROGRAM DON(OUTPUT)
                    CALL PRG1
                    CALL PRG2
                C   NOTE THIS IS A DUMMY PROGRAM
5               C   THIS DUMMY PROGRAM IS REQUIRED TO SET
                C   UP THE OUPUT BUFFER FOR THE FORTRAN
                C   OBJECT TIME ROUTINES.
                    END
```

4

```
 SUBROUTINE PRG2       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.43.19        PAGE     1

1                   SUBROUTINE PRG2
                    COMMON /ANSWERS/ NUMBERS(10)
                    PRINT *,(NUMBERS(I),I=1,10)
                    END
```

```
PRG1                                                   COMPASS 3.6-552.   81/10/07. 11.43.22.        PAGE     1

                                                 IDENT  PRG1
                                                 ENTRY  PRG1
                                                 LIST   -B,-R
    0  0000000000000000001      OPERAND  DATA   1,2,3,4,5,6,7,8,9,10
   12  0000000000000000000      PRG1     DATA   0
   13  6110000011                        SB1    9
   14  5111000000 +             LOOP     SA1    OPERAND+B1
                 10611                   BX6    X1
   15  5161000000 C                      SA6    SUM+B1
                 6111777776              SB1    B1-1
   16  0610000014 +                      PL     B1,LOOP
                 0400000012 +            EQ     PRG1
                                         USE    /ANSWERS/
    0                   12     SUM      BSS    10
   17                                    END

                437008 CM  STORAGE USED              15 STATEMENTS        4 SYMBOLS
                             SERIAL CPU ASSEMBLY          0.060 SECONDS          0 REFERENCES
```

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 6701

TRANSFER ADDRESS -- DON 2174

PROGRAM ENTRY POINTS -- DON 2174

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| DON | 111 | 2067 | LG02 | 81/10/07 | FTN | 4.8 552 | 666X I | PROGRAM OPT=1 |
| /ANSWERS/ | 2200 | 12 | | | | | | |
| PRG2 | 2212 | 11 | LG02 | 81/10/07 | FTN | 4.8 552 | 666X I | SUBROUTINEOPT=1 |
| PRG1 | 2223 | 17 | LG02 | 81/10/07 | COMPASS | 3.6 552 | | |
| SYSAID= | 2242 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2243 | 1 | | | | | | |
| /FCL.C./ | 2244 | 30 | | | | | | |
| /08.IO./ | 2274 | 130 | | | | | | |
| Q2NTRY= | 2424 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 552 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2424 | 43 | | | | | | |
| COMIO= | 2467 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2522 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FCL CAPSULE LOADING |
| FECMSK= | 2562 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2623 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2626 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON FLOATING OUTPUT CODE |
| FORSYS= | 3137 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 3640 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FCL MISC. UTILITIES. |
| GETFIT= | 3665 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| LDOUT= | 3761 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 4225 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON OUTPUT CODE |
| OUTF= | 4401 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 4605 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 4612 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 4652 | 14 | | | | | | |
| FDL.RES | 4666 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 5077 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 5321 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | 5756 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| ERR$RM | 6035 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 6062 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 6151 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 6156 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 6340 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 6346 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 6362 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 6420 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 6442 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | |
| CMM.R | 6451 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 6657 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |



.334 CP SECONDS  243008 CM STORAGE USED  5 TABLE MOVES

```
LOAD2,T5.
USER,DRF2755,DRF.
GET(TAPEIN=LOAD1)
LIBGEN(F=TAPEIN,P=USELIB1)
FTN,R=0.
LIBGEN(P=USELIB2)
REWIND,LGO.
FTN,R=0.
LIBGEN(P=USELIB3)
REPLACE(USELIB1,USELIB2,USFLIB3)
CATALOG(USELIB1,R,N,U)
CATALOG(USELIB2,R,N,U)
CATALOG(USELIB3,R,N,U)
R     7/8/9 END-OF-RECORD
      PROGRAM YOUMAIN(OUTPUT)
      PRINT 1
      CALL SUB1A
      CALL SUB2A
      CALL SUB3A
      CALL SUB4A
    1 FORMAT(1H1,*IN YOUMAIN*)
      END
      SUBROUTINE SUB1A
      PRINT *," IN SUB1A-PROGRAM YOUMAIN"
      END
      SUBROUTINE SUB2A
      PRINT *," IN SUB2A-PROGRAM YOUMAIN"
      END
      SUBROUTINE SUB3A
      PRINT *," IN SUB3A-PROGRAM YOUMAIN"
      END
      SUBROUTINE SUB4A
      PRINT *," IN SUB4A-PROGRAM YOUMAIN"
      END
R     7/8/9 END-OF-RECORD
      PROGRAM OURMAIN(OUTPUT)
      PRINT 1
      CALL SUB1B
      CALL SUB2B
      CALL SUB3B
      CALL SUB4B
      CALL SUB5B
    1 FORMAT(1H1,*IN OURMAIN*)
      END
      SUBROUTINE SUB1B
      PRINT *," IN SUB1B-PROGRAM OURMAIN"
      END
      SUBROUTINE SUB2B
      PRINT *," IN SUB2B-PROGRAM OURMAIN"
      END
      SUBROUTINE SUB3B
      PRINT *," IN SUB3B-PROGRAM OURMAIN"
      END
      SUBROUTINE SUB4B
      PRINT *," IN SUB4B-PROGRAM OURMAIN"
      END
      SUBROUTINE SUB5B
      PRINT *," IN SUB5B-PROGRAM OURMAIN"
      END
Q     6/7/8/9 END-OF-INFORMATION
```



```
      FULL  DAYFILE.  81/10/07. 11.44.11.*11.43.53* PAGE    1

11.43.53.LOAD2,T5.
11.43.53.UCCR, 64020,      0.057KCDS.
11.43.53.USER,DRF2755,.
11.43.53.GET(TAPEIN=LOAD1)
11.43.54.LIBGEN(F=TAPEIN,P=USELIB1)
11.43.54. LIBRARY GENERATION COMPLETE.
11.43.54.FTN,R=0.                                        1
11.43.59.       .126 CP SECONDS COMPILATION TIME
11.43.59.LIBGEN(P=USELIB2)
11.44.00. LIBRARY GENERATION COMPLETE.
11.44.00.REWIND,LGO.
11.44.00.FTN,R=0.                                        2
11.44.07.       .145 CP SECONDS COMPILATION TIME
11.44.07.LIBGEN(P=USELIB3)
11.44.08. LIBRARY GENERATION COMPLETE.
11.44.08.REPLACE(USELIB1,USELIB2,USELIB3)
11.44.10.CATALOG(USELIB1,R,N,U)
11.44.10. CATALOG COMPLETE.                              3
11.44.10.CATALOG(USELIB2,R,N,U)
11.44.10. CATALOG COMPLETE.                                4
11.44.10.CATALOG(USELIB3,R,N,U)                              5
11.44.11. CATALOG COMPLETE.
11.44.11.UEAD,       0.001KUNS.
11.44.11.UEPF,       0.032KUNS.
11.44.11.UEMS,       6.424KUNS.
11.44.11.UECP,       0.402SECS.
11.44.11.AESR,       2.897UNTS.
11.44.11.$OUT(*/DP=E)
11.44.11. NO FILES PROCESSED.
11.44.11.$DAYFILE(OUTPUT,JT=D)
11.45.24.UCLP, 64,023,      1.216KLNS.
```

```
    PROGRAM YOUMAIN     73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.43.54        PAGE     1

1                   PROGRAM YOUMAIN(OUTPUT)
                    PRINT 1
                    CALL SUB1A
                    CALL SUB2A
5                   CALL SUB3A
                    CALL SUB4A
                  1 FORMAT(1H1,*IN YOUMAIN*)
                    END
```

1

```
 SUBROUTINE SUB1A       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.43.54        PAGE     1

1                   SUBROUTINE SUB1A
                    PRINT *,# IN SUB1A-PROGRAM YOUMAIN#
                    END
```

```
 SUBROUTINE SUB2A       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.43.54        PAGE     1

1                   SUBROUTINE SUB2A
                    PRINT *,# IN SUB2A-PROGRAM YOUMAIN#
                    END
```

```
 SUBROUTINE SUB3A       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.43.54        PAGE     1

1                   SUBROUTINE SUB3A
                    PRINT *,# IN SUB3A-PROGRAM YOUMAIN#
                    END
```

```
 SUBROUTINE SUB4A       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.43.54        PAGE     1

1                   SUBROUTINE SUB4A
                    PRINT *,# IN SUB4A-PROGRAM YOUMAIN#
                    END
```

```
   PROGRAM OURMAIN     73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.44.00        PAGE      1

1                PROGRAM OURMAIN(OUTPUT)
                 PRINT 1
                 CALL SUB1B
                 CALL SUB2B
5                CALL SUB3B
                 CALL SUB4B
                 CALL SUB5B
               1 FORMAT(1H1,*IN OURMAIN*)
                 END
```

2

```
 SUBROUTINE SUB1B      73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.44.00        PAGE      1

1                SUBROUTINE SUB1B
                 PRINT *,# IN SUB1B-PROGRAM OURMAIN#
                 END
```

```
 SUBROUTINE SUB2B      73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.44.00        PAGE      1

1                SUBROUTINE SUB2B
                 PRINT *,# IN SUB2B-PROGRAM OURMAIN#
                 END
```

```
 SUBROUTINE SUB3B      73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.44.00        PAGE      1

1                SUBROUTINE SUB3B
                 PRINT *,# IN SUB3B-PROGRAM OURMAIN#
                 END
```

```
 SUBROUTINE SUB4B      73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.44.00        PAGE      1

1                SUBROUTINE SUB4B
                 PRINT *,# IN SUB4B-PROGRAM OURMAIN#
                 END
```

```
 SUBROUTINE SUB5B      73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.44.00        PAGE      1

1                SUBROUTINE SUB5B
                 PRINT *,# IN SUB5B-PROGRAM OURMAIN#
                 END
```

```
     CATALOG OF USELIB1          FILE     1                    81/10/07. 11.44.10.      PAGE      1
REC  NAME      TYPE        LENGTH     CKSUM     DATE     COMMENTS

  1  USELIB1   ULIB            40      4654  81/10/07.
  2  MAIN      REL            112      0021  81/10/07. PROGRAM    OPT=1
       MAIN
       OUTPUT=
  3  SUB1      REL             40      6615  81/10/07. SUBROUTINEOPT=1
       SUB1
  4  SUB2      REL             40      4201  81/10/07. SUBROUTINEOPT=1
       SUB2
  5  SUB3      REL             40      1545  81/10/07. SUBROUTINEOPT=1
       SUB3
  6  SUB4      REL             40      7131  81/10/07. SUBROUTINEOPT=1
       SUB4
  7  USELIB1   OPLD            15      3005  81/10/07.

  8  * EOF *       SUM =      367
```

3



```
     CATALOG OF USELIB1          FILE     2                    81/10/07. 11.44.10.      PAGE      2
REC  NAME      TYPE        LENGTH     CKSUM     DATE     COMMENTS

       * EOI *       SUM =        0
```

CATALOG OF USELIB2 FILE 1 81/10/07. 11.44.10. PAGE 1

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | USELIB2 | ULIB | 40 | 2500 | 81/10/07. | |
| 2 | YOUMAIN<br>YOUMAIN<br>OUTPUT= | REL | 112 | 7441 | 81/10/07. | PROGRAM OPT=1 |
| 3 | SUB1A<br>SUB1A | REL | 40 | 2501 | 81/10/07. | SUBROUTINEOPT=1 |
| 4 | SUB2A<br>SUB2A | REL | 40 | 0115 | 81/10/07. | SUBROUTINEOPT=1 |
| 5 | SUB3A<br>SUB3A | REL | 40 | 5451 | 81/10/07. | SUBROUTINEOPT=1 |
| 6 | SUB4A<br>SUB4A | REL | 40 | 3065 | 81/10/07. | SUBROUTINEOPT=1 |
| 7 | USELIB2 | OPLD | 15 | 4617 | 81/10/07. | |
| 8 | * EOF * | SUM = | 367 | | | |

4

CATALOG OF USELIB2 FILE 2 81/10/07. 11.44.10. PAGE 2

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| | * EOI * | SUM = | 0 | | | |

CATALOG OF USELIB3 FILE 1 81/10/07. 11.44.10. PAGE 1

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | USELIB3 | ULIB | 54 | 5045 | 81/10/07. | |
| 2 | OURMAIN<br>OURMAIN<br>OUTPUT= | REL | 114 | 5707 | 81/10/07. | PROGRAM OPT=1 |
| 3 | SUB1B<br>SUB1B | REL | 40 | 5560 | 81/10/07. | SUBROUTINEOPT=1 |
| 4 | SUB2B<br>SUB2B | REL | 40 | 7204 | 81/10/07. | SUBROUTINEOPT=1 |
| 5 | SUB3B<br>SUB3B | REL | 40 | 0530 | 81/10/07. | SUBROUTINEOPT=1 |
| 6 | SUB4B<br>SUB4B | REL | 40 | 2154 | 81/10/07. | SUBROUTINEOPT=1 |
| 7 | SUB5B<br>SUB5B | REL | 40 | 7150 | 81/10/07. | SUBROUTINEOPT=1 |
| 8 | USELIB3 | OPLD | 17 | 7265 | 81/10/07. | |
| 9 | * EOF * | SUM = | 447 | | | |

5

CATALOG OF USELIB3 FILE 2 81/10/07. 11.44.10. PAGE 2

| REC | NAME | TYPE | LENGTH | CKSUM | DATE | COMMENTS |
|---|---|---|---|---|---|---|
| | * EOI * | SUM = | 0 | | | |

## EXERCISE 11

PROBLEM (The libraries built in this example will be used in the exercise that follows the loader chapter).

1. Access the source program library containing MAIN, SUB1 and SUB2, extract and compile these programs and execute them twice.

2. (Required) Access the same library but only extract SUB1 and SUB2, compile them, and create a user library of these two subroutines and make a permanent file of this library.

3. Create another library which contains SUB1 and SUB2, and make it a permanent file.

# LESSON 7
# JOB CONTROL LANGUAGE

LESSON PREVIEW:

In this lesson, the student will learn how to structure a job by the use of CYBER Control Language (CCL).

OBJECTIVES:

After completing this lesson the student should be able to -

- State the rules of syntax governing formation of jobs using CYBER Control Language.
- Generate procedure files.
- Call procedure files.

TRAINING AIDS:

None.

PROJECTS:

Exercise 12 (problem).

REFERENCES:

NOS Version 1 Reference Manual, Volume 1, Section 4.

## MAJOR CCL COMMANDS AND FEATURES

### PROCEDURE FILES

.PROC,PNAME*I,$P_1$,$P_2$,...,$P_N$.

WHERE $N \leq 50$

BEGIN,PNAME,FNAME,$P_1$,$P_2$,...,$P_N$.

REVERT
.HELP
.ENDHELP

### CONTROL FLOW

IFE, EXP, LABEL.
ELSE, LABEL.
ENDIF, LABEL.
SKIP, LABEL.
WHILE, LABEL.
ENDW, LABEL.
GOTO, LABEL.

### USER-SET VARIABLES

R1, R2, R3 — SET, R1 = R1+1,
RIG
SW1, SW2, ..., SW6
SS
DSC

## SYSTEM-SET VARIABLES & CONSTANTS

| | | |
|---|---|---|
| OT | ORIGIN TYPE | COMPARE BCO, EIO, TXO, SYO. |
| PNL | NESTING LEVEL | |
| TIME | | |
| EF | ERROR FLAG | COMPARE TLE, CPE, MNE, ODE. |

## EXPRESSIONS

OPERATORS: +, -, /, *, **, =, , ,
.EQ., .LT., .GT., .NE., .LE., .GE.,
.AND., .OR., .NOT.

OPERANDS: BINARY OR DECIMAL INTEGERS UP TO 10 DIGITS
SYMBOLIC NAMES RECOGNIZED BY CCL
CCL FUNCTIONS AND EXPRESSIONS

## SPECIAL COMMANDS

DISPLAY, EXP

| | |
|---|---|
| .DATA, LFN | CREATES A LOCAL DATA FILE |
| .EOR | INSERTS END OF RECORD MARK IN THE PROCEDURE COMMAND RECORD OR A DATA FILE CREATED BY THE PROCEDURE |
| .EOF | SIMILAR TO EOR EXCEPT INSERT END OF FILE |
| .* | COMMENT |

## FILE FUNCTION

FILE (LFN, XXX)

'XXX' IS CCL SYMBOLIC NAME FOR VARIOUS TYPES OF FILE ATTRIBUTES. LIST INCLUDES:

| | | |
|---|---|---|
| MS | AS | BOI |
| TP | LO | EOI |
| TT | PF | WR |

## FILE FUNCTION EXAMPLES:

IFE, FILE (MYFILE, AS), L1.

TEST TO SEE IF MYFILE IS KNOWN TO THE JOB.
IF MYFILE NOT KNOWN, JUMP TO L1.

IFE, FILE (BIGFILE, PR.AND.WR), OK.

TEST TO SEE IF BIGFILE IS A DIRECT ACCESS PERMANENT FILE, CURRENTLY ATTACHED TO THE JOB WITH WRITE PERMISSION ENABLED.
IF NOT, JUMP TO OK.

CYBER CONTROL LANGUAGE IS DESCRIBED IN SECTION 1-4 OF THE NOS REFERENCE MANUAL.

## .PROC Directive.

Procedures provide the capabilities to validate the procedure parameters entered by the user, both automatically generated and application programmer provided, provide help for the procedure and for each parameter, and provide an interactive prompting dialogue with the user either at the user's request, or when the user makes an error in entering a parameter.

The format of the .PROC directive

A: .PROC,PNAME,$P_1$,$P_2$,...,$P_N$.

NO PARAMETER PROMPTING

B: .PROC,PNAME*M,PN["TITLE"]=("PROMPT",,)

Defines - Menu procedure. A menu procedure displays the choices available and prompts for a unique selection. The selection will control conditional interpretation or execution of the body of the procedure.

C: .PROC,PNAME*I,$P_1$,$P_2$,...PN

Enables parameter prompting.

WHERE FORMAT OF THE PI IS:

KEYWRD "DESCRIPTION" = (CHECKLIST)

KEYWRD SPECIFIES THE KEYWORD OF THE PARAMETER.

DESCRIPTION SPECIFIES TEXT STRING THAT SYSTEM DISPLAYS WHEN PROMPTING.

CHECKLIST SPECIFIES ACCEPTABLE VALUES, DEFAULT VALUES, SYNTAX AND REQUIRED VALUES. THE POSSIBLE VALUES IN CHECKLIST ARE

*N, *F, *K, *A, *SN

IF *N DOESN'T APPEAR IN CHECKLIST, PARAMETER IS REQUIRED.

IF *F APPEARS IN CHECKLIST, PARAMETER MUST CONFORM TO THE OPERATING SYSTEM FORMAT FOR A LOCAL FILE NAME.

EXAMPLE:

PROCEDURE PRINT VERIFIES THAT THE SELECTION FILE IS LOCAL BEFORE IT ROUTES THE FILE TO A PRINTER.

```
.PROC,PRINT*I,F"FILE NAME"=(*F),
DC=(*N=LP,LP,PR,LR,LS,LT).
.HELP
THIS PROCEDURE ROUTES A PERMANENT FILE TO THE SELECTED LINE PRINTER.
.HELP,F
THE NAME OF THE PERMANENT FILE TO BE ROUTED.
.HELP,DC
THE DISPOSITION CODE.  DC ACCEPTS ONLY THE LINE PRINTER OPTIONS.
.ENDHELP
IFE,FILE (F,,NOT,LO),PF.
GET,F.
ELSE,PF.
REWIND,F.
ENDIF,PF.
COPYSBF,F,Z.
ROUTE,Z,#DC=DC.
REVERT,NOLIST.
```

NOTE: PARAMETER F IS REQUIRED SINCE NO FORM OF *N EXISTS IN CHECKLIST AND F MUST CONFORM TO A LOCAL FILE NAME SINCE *F FIRST IN CHECKLIST.

PARAMETER DC IS NOT REQUIRED AND IF NO VALUE SPECIFIED, LP WILL BE USED.

IF PROCEDURE PRINT IS ON LOCAL FILE CALLED PRINT,
TO GET DESCRIPTION ENTER ANY OF THE FOLLOWING:

```
PRINT?
PRINT,?
BEGIN, PRINT, PRINT?
BEGIN,, PRINT?
```

THE FOLLOWING IS A SAMPLE OF THE DIALOGUE GENERATED
BY THE .PROC AND .HELP DIRECTIVES:

```
BEGIN,PRINT,PRINT?
THIS PROCEDURE ROUTES A PERMANENT FILE TO THE SELECTED LINE PR
PARAMETERS FOR PRINT ARE F, DC
ENTER F FILE NAME? F?
ALLOWABLE VALUE(S)
MUST BE A FILE NAME
THE NAME OF THE PERMANENT FILE TO BE ROUTED.

ENTER F FILE NAME?  DATAFIL
ENTER DC ? DC?
ALLOWABLE VALUE(S)
   LP
   PR
   LR
   LS
   LT
PARAMETER MAY BE OMITTED
THE DISPOSITION CODE. DC ACCEPTS ONLY THE LINE PRINTER OPTIONS

ENTER DC ? PR
```

## PROCEDURE FILE EXAMPLES

1) TASK - DESIGN A PROCEDURE TO MAKE 20 COPIES OF A FILE. USE COPYSBF TO ENSURE THAT A BLANK IS USED FOR THE CARRIAGE CONTROL IF THE FILE IS SENT TO A LINE PRINTER.

```
.PROC,DOIT.
SET,R1=0.
WHILE,R1.LT.20,LOOP.
COPYSBF,SAMPLE,TEMP.
REWIND,SAMPLE.
SET,R1=R1+1.
ENDW,LOOP.
```

THE PROCEDURE MAY BE INVOKED BY EXECUTING A

```
BEGIN,DOIT.
```

THIS ASSUMES THAT DOIT RESIDES ON THE DEFAULT PROCEDURE FILE (PROCFIL).

2) TASK - GENERALIZE THE PROCEDURE TO HANDLE ANY LFN AND GENERATE ANY NUMBER OF COPIES.

```
.PROC,DOIT,LFN,NUM=1.
SET,R1=0.
WHILE,R1.LT.NUM,LOOP.
REWIND,LFN.
COPYSBF,LFN,TEMP.
SET,R1=R1+1.
ENDW,LOOP.
```

THE PROCEDURE MAY BE INVOKED BY EXECUTING A

```
BEGIN,DOIT,,FILE1,5.
```

NOTE THE DOUBLE COMMA IN THE BEGIN STATEMENT - USES THE DEFAULT PROCEDURE FILE NAME. NOTE THE DEFAULT REPEAT COUNT (NUM) ON THE PROCEDURE HEADER.

3) TASK - IF THE LFN IS NOT CURRENTLY ASSIGNED TO THIS JOB, ATTACH THE LFN.

```
.PROC,DOIT,LFN,NUM=1.
REWIND,TEMP.
IFE,.NOT.(FILE(LFN,AS)),GOTIT.
ATTACH,LFN.
ENDIF,GOTIT.
SET,R1=0.
WHILE,R1.LT.NUM,LOOP.
REWIND,LFN.
COPYSBF,LFN,TEMP.
SET,R1=R1+1.
ENDW,LOOP.
REWIND,LFN,TEMP.
```

THE PROCEDURE MAY BE INVOKED BY EXECUTING A

```
BEGIN,DOIT,,FILE1,10.
```

4) TASK - AUTOMATICALLY ROUTE THE FILE TO THE LINE PRINTER. ALSO, ALLOW USER TO SPECIFY ALTERNATE USER NUMBER FOR PERMANENT FILE.

```
.PROC,DOIT,LFN,NUM=1,UN=FOX0421.
REWIND,TEMP.
IFE,.NOT.(FILE(LFN,AS)),GOTIT.
ATTACH,LFN/≡UN=UN.
ENDIF,GOTIT.
SET,R1=0.
WHILE,R1.LT.NUM,LOOP.
REWIND,LFN.
COPYSBF,LFN,TEMP.
SET,R1=R1=1.
ENDW,LOOP.
ROUTE,TEMP,DC=PR.
REWIND,LFN.
```

THE PROCEDURE MAY BE INVOKED BY EXECUTING A

```
BEGIN,DOIT,,FILE4,6,GAD0635.
```

NOTE THE USE OF THE ≡ TO SUPPRESS SUBSTITUTION.

5) TASK - MAKE COPYSBF OPTIONAL FOR FILES THAT ALREADY HAVE PROPER CARRIAGE CONTROLS.

```
.PROC,DOIT,LFN,NUM=1,UN=FOX0421,CC=.
REWIND,TEMP.
IFE,.NOT.(FILE(LFN,AS)),GOTIT.
ATTACH,LFN/=UN=UN.
ENDIF,GOTIT.
SET,R1=0.
WHILE,R1.LT.NUM,LOOP.
REWIND,LFN.
COPY→ CC,LFN,TEMP.
SET,R1=R1+1.
ENDW,LOOP.
ROUTE,TEMP,DC=PR.
REWIND,LFN.
```

THE PROCEDURE MAY BE INVOKED BY EXECUTING A

```
BEGIN,DOIT,,FILE5,7,XYZ1357,SBF.
```

NOTE THE USE OF THE → FOR SUBSTITUTION WITHIN A STRING.

A USER MAY DEFINE A PROCEDURE WITH NO PARAMETERS THAT WILL BE EXECUTED EVERY TIME HE LOGS INTO THE SYSTEM. THE UPROC COMMAND SPECIFIES A PERMANENT INDIRECT ACCESS FILE IN THE USER'S CATALOG THAT WILL CONTAIN THE PROCEDURE TO BE EXECUTED.

3

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 7243

TRANSFER ADDRESS -- ADD 2174

PROGRAM ENTRY POINTS -- ADD 2174

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| ADD | 111 | 2113 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | PROGRAM OPT=1 |
| SYSAID= | 2224 | 1 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2225 | 1 | | | | | | | |
| /FCL.C./ | 2226 | 27 | | | | | | | |
| /Q8.IO./ | 2255 | 127 | | | | | | | |
| Q2NTRY= | 2404 | 0 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2404 | 41 | | | | | | | |
| COMIO= | 2445 | 33 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2500 | 40 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL CAPSULE LOADING |
| FECMSK= | 2540 | 41 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2601 | 3 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2604 | 311 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON FLOATING OUTPUT CODE |
| FMTAP= | 3115 | 357 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | 3474 | 376 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 4072 | 25 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL MISC. UTILITIES. |
| GETFIT= | 4117 | 74 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | 4213 | 465 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | OUTPUT FORMAT INTERPRETER. |
| OUTCOM= | 4700 | 154 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON OUTPUT CODE |
| OUTC= | 5054 | 152 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FORMATTED WRITE FORTRAN RECORD. |
| CPU.CPM | 5226 | 5 | SL-SYSLIB | 81/09/18 | COMPASS | 3.6 | 552 | | 79/05/10. 81/08/07. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 5233 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 5273 | 14 | | | | | | | |
| FDL.RES | 5307 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 5520 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 5742 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| ERR$RM | 6377 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 6424 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 6513 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 6520 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 6702 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 6710 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 6724 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 6762 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 7004 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | 7013 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 7221 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

.337 CP SECONDS 245008 CM STORAGE USED 3 TABLE MOVES
5050 50

```
CCL7,T5.
USER,DRF2755,DRF.
COPYBR,,DOG.
REWIND,DOG)
MAP(PART)
BEGIN,FTNMOD,CCLEX5,USERMOD=DOG.
CATLIST.
PURGE(CCLEX1,CCLEX2,CCLEX3,CCLEX4,CCLEX5,CCLEX6)
CATLIST.
R     7/8/9 END-OF-RECORD
*IDENT MOD2
*INSERT ADD.4
       N=50
*DELETE ADD.5
       DO 10 I=1,N
Q     6/7/8/9 END-OF-INFORMATION
```

```
                    FULL  DAYFILE.  81/10/01. 11.44.24.*11.44.18* PAGE    1

11.44.18.CCL7,T5.
11.44.18.UCCR, 64020,       0.014KCDS.
11.44.18.USER,DRF2755,.
11.44.18.COPYBR,,DOG.
11.44.18. COPY COMPLETE.
11.44.18.REWIND,DOG)
11.44.18.MAP(PART)
11.44.18.BEGIN,FTNMOD,CCLEX5,USERMOD=DOG.
11.44.19.GET(OLDPL=CCLEX4)
11.44.19.UPDATE(I=ZZCCLAA,F)                                    1
11.44.19. UPDATE COMPLETE.
11.44.20.IFE,(0.NE.0),IF1.
11.44.20.ENDIF,IF1.
11.44.20.FTN(I,R=0)
11.44.21.       .087 CP SECONDS COMPILATION TIME                2
11.44.21.IFE,(0.NE.0),IF2.
11.44.21.ENDIF,IF2.
11.44.21.LGO.
11.44.22.      END  ADD                                         3
11.44.22.     013500 MAXIMUM EXECUTION FL.
11.44.22.       .037 CP SECONDS EXECUTION TIME.
11.44.22.$REVERT.CCL
11.44.23.CATLIST.                                               4
11.44.23. CATLIST COMPLETE.
11.44.23.PURGE(CCLEX1,CCLEX2,CCLEX3,CCLEX4,CCLEX5,CCLEX6)
11.44.23.CATLIST.
11.44.24. CATLIST COMPLETE.                                     5
11.44.24.UEAD,       0.001KUNS.
11.44.24.UEPF,       0.050KUNS.
11.44.24.UEMS,       1.712KUNS.
11.44.24.UECP,       0.585SECS.
11.44.24.AESR,       1.981UNTS.
11.44.24.$OUT(*/OP=E)
11.44.24. NO FILES PROCESSED.
11.44.24.$DAYFILE(OUTPUT,JT=0)
11.44.34.UCLP, 64,023,      0.512KLNS.
```

```
UNLABELED OLDPL                IDENT    MOD1                    UPDATE 1.4-528.      81/10/01. 11.44.19.      PAGE    1

*****     *IDENT MOD1
*****     *DELETE ADD.7
                IAVE=ISUM/I
                WRITE(2,20) ISUM,IAVE
             20 FORMAT(10X,2I5)
*****     *READ DOG                                                                                    DOG
```

1

```
UNLABELED OLDPL                IDENT    MOD2                    UPDATE 1.4-528.      81/10/01. 11.44.19.      PAGE    2

*****     *IDENT MOD2                                                                                  DOG
*****     *INSERT ADD.4                                                                                DOG
                N=50                                                                                   DOG
*****     *DELETE ADD.5                                                                                DOG
                DO 10 I=1,N                                                                            DOG
```

MODIFICATIONS / CONTROL CARDS

```
ADD             N=50                                                  MOD2          1      I
ADD             DO 10 I=1,100                                         ADD           5        D
ADD             DO 10 I=1,N                                           MOD2          2      I
ADD             WRITE(2,*) ISUM                                       ADD           7        D
ADD             IAVE=ISUM/I                                           MOD1          1      I
ADD             WRITE(2,20) ISUM,IAVE                                 MOD1          2      I
ADD          20 FORMAT(10X,2I5)                                       MOD1          3      I
```

CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

ADD MOD1 MOD2

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANK$$$ ADD

DECKS WRITTEN TO COMPILE FILE

ADD

THIS UPDATE REQUIRED 37000B WORDS OF CORE.

```
      PROGRAM ADD        73/74   OPT=1                    FTN 4.8+528        81/10/01. 11.44.20        PAGE      1

                                                                                                          2
  1                 PROGRAM ADD(OUTPUT,TAPE2=OUTPUT)                                    ADD         2
          C         USE A DO LOOP TO SUM THE NUMBERS FROM 1 TO 100                      ADD         3
                    ISUM=0                                                              ADD         4
                    N=50                                                                MOD2        1
  5                 DO 10 I=1,N                                                         MOD2        2
             10     ISUM=ISUM+I                                                         ADD         6
                    IAVE=ISUM/I                                                         MOD1        1
                    WRITE(2,20) ISUM,IAVE                                               MOD1        2
             20     FORMAT(10X,2I5)                                                     MOD1        3
 10                 END                                                                 ADD         8
```

3

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 7245

TRANSFER ADDRESS -- ADD 2174

PROGRAM ENTRY POINTS -- ADD 2174

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| ADD | 111 | 2115 | LGO | 81/10/01 | FTN | 4.8 | 528 | 666X I | PROGRAM OPT=1 |
| SYSAID= | 2226 | 1 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2227 | 1 | | | | | | | |
| /FCL.C./ | 2230 | 27 | | | | | | | |
| /99.IO./ | 2257 | 127 | | | | | | | |
| O2NTRY= | 2406 | 0 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2406 | 41 | | | | | | | |
| COMIO= | 2447 | 33 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2502 | 40 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL CAPSULE LOADING |
| FECMSK= | 2542 | 41 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2603 | 3 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2606 | 311 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON FLOATING OUTPUT CODE |
| FMTAP= | 3117 | 357 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | 3476 | 376 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 4074 | 25 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FCL MISC. UTILITIES. |
| GETFIT= | 4121 | 74 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | 4215 | 465 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | OUTPUT FORMAT INTERPRETER. |
| OUTCOM= | 4702 | 154 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | COMMON OUTPUT CODE |
| OUTC= | 5056 | 152 | SL-FORTRAN | 80/09/25 | COMPASS | 3.6 | 528 | | FORMATTED WRITE FORTRAN RECORD. |
| CPU.CPM | 5230 | 5 | SL-SYSLIB | 81/09/18 | COMPASS | 3.6 | 552 | | 79/05/10. 81/08/07. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 5235 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 5275 | 14 | | | | | | | |
| FDL.RES | 5311 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 5522 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 5744 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| ERR$RM | 6401 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 6426 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 6515 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 6522 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 6704 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 6712 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 6726 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 6764 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 7006 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | 7015 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 7223 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

.320 CP SECONDS 24500B CM STORAGE USED 3 TABLE MOVES
1275 25

```
CATALOG OF  DRF2755          FM/FEATURE 81/10/01. 11.44.23.  PAGE   1

INDIRECT ACCESS FILE(S)

 CCLEX1     CCLEX4     DATAPL1    LOAD1      TESTCOP    UPDATED    USELIB2
 CCLEX2     CCLEX5     DEMO       SEG        UPDATEC    USELIB1    USELIB3
 CCLEX3     CCLEX6

DIRECT ACCESS FILE(S)

 CKPFL

       16 INDIRECT ACCESS FILE(S),    TOTAL PRUS =       263.
        1 DIRECT ACCESS FILE(S),      TOTAL PRUS =         0.
```

4



```
CATALOG OF  DRF2755          FM/FEATURE 81/10/01. 11.44.24.  PAGE   1

INDIRECT ACCESS FILE(S)

 DATAPL1    LOAD1      TESTCOP    UPDATED    USELIB1    USELIB2    USELIB3
 DEMO       SEG        UPDATEC

DIRECT ACCESS FILE(S)

 CKPFL

       10 INDIRECT ACCESS FILE(S),    TOTAL PRUS =       250.
        1 DIRECT ACCESS FILE(S),      TOTAL PRUS =         0.
```

5

```
CCL8,T5.
USER,DRF2755,DRF.
COPYBR,,PROCD.
BEGIN,FTNFIL,PROCD.
COMMENT. CCL PROCEDURE RETURNED HERE
R
.PROC,FTNFIL.
COMMENT. THIS IS FIRST MESSAGE
IFE,EF=0,L1.
COMMENT. THIS IS SECOND MESSAGE
IFE,EF=0,L2.
REVERT.
ENDIF,L2.
DISPLAY(EF)
REVERT.
ENDIF,L1.
CATLIST.
LIMITS.
R
Q
```

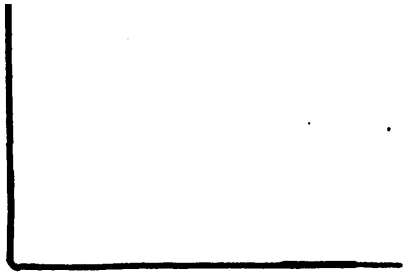
```
    FULL  DAYFILE.  81/10/01. 11.45.13.*11.45.11* PAGE    1

11.45.11.CCL8,T5.
11.45.11.UCCR, 64020,      0.017KCDS.
11.45.11.USER,DRF2755,.
11.45.11.COPYBR,,PROCD.
11.45.11. COPY COMPLETE.
11.45.12.BEGIN,FTNFIL,PROCD.
11.45.12.COMMENT. THIS IS FIRST MESSAGE
11.45.12.IFE,EF=0,L1.
11.45.12.COMMENT. THIS IS SECOND MESSAGE
11.45.12.IFE,EF=0,L2.
11.45.12.REVERT.
11.45.12.COMMENT. CCL PROCEDURE RETURNED HERE
11.45.12.UEAD,       0.001KUNS.
11.45.12.UFMS,       0.227KUNS.
11.45.12.UECP,       0.041SECS.
11.45.12.AESR,       1.071UNTS.
11.45.12.$OUT(*/OP=E)
11.45.12. NO FILES PROCESSED.
11.45.13.$DAYFILE(OUTPUT,JT=D)
11.45.16.UCLP, 64,023,      0.128KLNS.
```

```
CCL9.    CONVERT THIS N.C.L. TO C.C.L.
USER,DRF2755,DRF.
COPYBR,,DUP.
COPYBR,,RID.
*
CALL,DUP(DUPFILE=FIRST)
RESEQ(FIRST)
CALL,RID(RIDFILE=FIRST)  OUTPUT NUMBER 1
*
CALL,DUP(DUPFILE=SEC)
PACK,SEC.
CALL,RID(RIDFILE=SEC)    OUTPUT NUMBER 2
*
CALL,DUP(DUPFILE=THREE)
SORT(THREE)
CALL,RID(RIDFILE=THREE)  OUTPUT NUMBER 3
R  7/8/9 END-OF-RECORD
DUP
*
* THIS PROC CREATES A TWO-RECORD FILE FROM THE    LAST TWO RECORDS OF THIS
*
REWIND(INPUT)
SKIPR(INPUT,3)
COPY(INPUT,DUPFILE)
REWIND(DUPFILE)
R  7/8/9 END-OF-RECORD
RID
*
* THIS PROCEDURE SENDS THE RESULTS TO THE PRINTER  FOR PRINTING.
*
REWIND(RIDFILE)
COPY(RIDFILE,OUTPUT)
OUT.
R  7/8/9 END-OF-RECORD
9100 FIRST LINE - ORIGIONAL
9110 SECOND LINE -ORIGIONAL
9120 THIRD LINE - ORIGIONAL
R  7/8/9 END-OF-RECORD
9100 FIRST LINE - FIRST CHANGE
9115 INSERT BETWEEN SECOND AND THIRD LINES
9110
Q  6/7/8/9 END-OF-INFORMATION
```





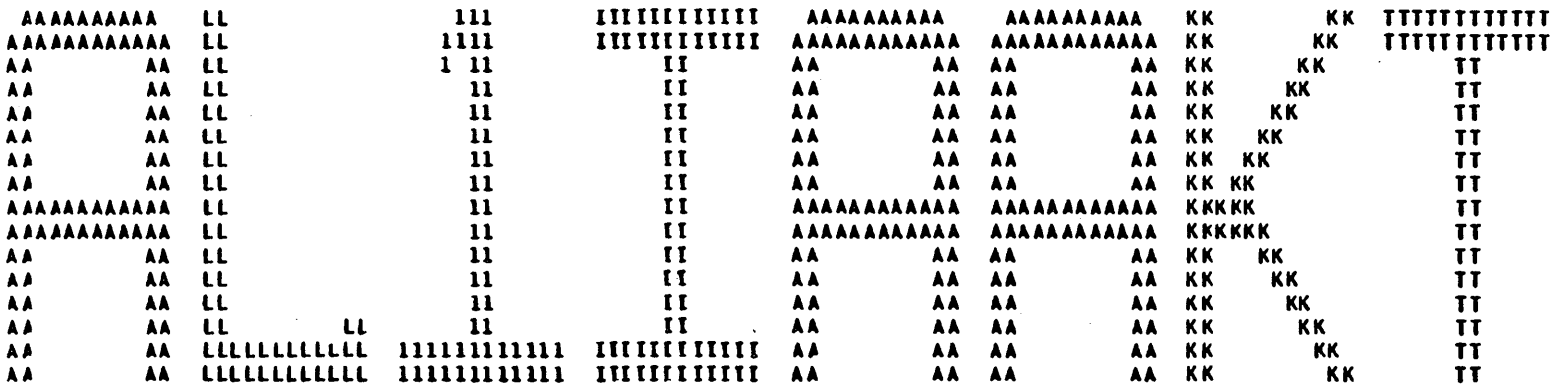
```
11.47.14.CCL9.    CONVERT THIS N.C.L. TO C.C.L.
11.47.14.UCCR, 64020,      0.037KCDS.
11.47.14.USER,DRF2755,.
11.47.14.COPYBR,,DUP.
11.47.14. COPY COMPLETE.
11.47.14.COPYBR,,RID.
11.47.15. COPY COMPLETE.
11.47.15.*
11.47.15.CALL,DUP(DUPFILE=FIRST)
11.47.15.*
11.47.15.* THIS PROC CREATES A TWO-RECORD FILE FROM THE
11.47.15. LAST TWO RECORDS OF THIS DECK
11.47.15.*
11.47.15.REWIND(INPUT)
11.47.15.SKIPR(INPUT,3)
11.47.15.COPY(INPUT,FIRST)
11.47.15. EOI ENCOUNTERED.
11.47.15.REWIND(FIRST)
11.47.15.RESEQ(FIRST)
11.47.15.CALL,RID(RIDFILE=FIRST)  OUTPUT NUMBER 1
11.47.16.*
11.47.16.* THIS PROCEDURE SENDS THE RESULTS TO THE PRINTER
11.47.16. FOR PRINTING.
11.47.16.*
11.47.16.REWIND(FIRST)
11.47.16.COPY(FIRST,OUTPUT)
11.47.16. EOI ENCOUNTERED.
11.47.16.OUT.
11.47.16. 1 FILE PROCESSED.
11.47.16.*
11.47.16.CALL,DUP(DUPFILE=SEC)
11.47.17.*
11.47.17.* THIS PROC CREATES A TWO-RECORD FILE FROM THE
11.47.17. LAST TWO RECORDS OF THIS DECK
11.47.17.*
11.47.17.REWIND(INPUT)
11.47.17.SKIPR(INPUT,3)
11.47.17.COPY(INPUT,SEC)
11.47.17. EOI ENCOUNTERED.
11.47.17.REWIND(SEC)
11.47.17.PACK,SEC.
11.47.17. PACK COMPLETE.
11.47.17.CALL,RID(RIDFILE=SEC)    OUTPUT NUMBER 2
11.47.17.*
11.47.17.* THIS PROCEDURE SENDS THE RESULTS TO THE PRINTER
11.47.17. FOR PRINTING.
11.47.18.*
11.47.18.REWIND(SEC)
11.47.18.COPY(SEC,OUTPUT)
11.47.18. EOI ENCOUNTERED.
11.47.18.OUT.
11.47.18. 1 FILE PROCESSED.
11.47.18.*
11.47.18.CALL,DUP(DUPFILE=THREE)
11.47.18.*
11.47.18.* THIS PROC CREATES A TWO-RECORD FILE FROM THE
11.47.18. LAST TWO RECORDS OF THIS DECK
11.47.18.*
```



```
FULL  DAYFILE.  81/10/07. 11.47.19.*11.47.18* PAGE    2

11.47.18.REWIND(INPUT)
11.47.18.SKIPR(INPUT,3)
11.47.18.COPY(INPUT,THREE)
11.47.18. EOI ENCOUNTERED.
11.47.18.REWIND(THREE)
11.47.18.SORT(THREE)
11.47.19.CALL,RID(RIDFILE=THREE)  OUTPUT NUMBER 3
11.47.19.*
11.47.19.* THIS PROCEDURE SENDS THE RESULTS TO THE PRINTER
11.47.19. FOR PRINTING.
11.47.19.*
11.47.19.REWIND(THREE)
11.47.19.COPY(THREE,OUTPUT)
11.47.19. EOI ENCOUNTERED.
11.47.19.OUT.
11.47.19. ERROR IN ROUTE FUNCTION, LFN = OUTPUT.
11.47.19. DSP - TOO MANY DEFERRED BATCH JOBS.
11.47.19.UEAD,       0.001KUNS.
11.47.19.UEMS,       0.513KUNS.
11.47.19.UECP,       0.219SECS.
11.47.19.AESR,       1.286UNTS.
11.47.19.$OUT(*/DP=E)
11.47.19. NO FILES PROCESSED.
11.47.19.$DAYFILE(OUTPUT,JT=D)
11.47.28.UCLP, 64,023,       0.192KLNS.
```

OPERATING SYSTEM = NOS 4K30-FNT101 81/10/05. PRINTED = 81/10/07. 11.47.17.

UJN = CCL9 FAMILY = FEATURE JOB ORIGIN = BATCH.
CREATING JSN = AAKS USER NUMBER = DRF2755 SERVICE CLASS = BATCH.




7-22

1

0100 FIRST LINE - ORIGIONAL

0110 SECOND LINE -ORIGIONAL

0120 THIRD LINE - ORIGIONAL

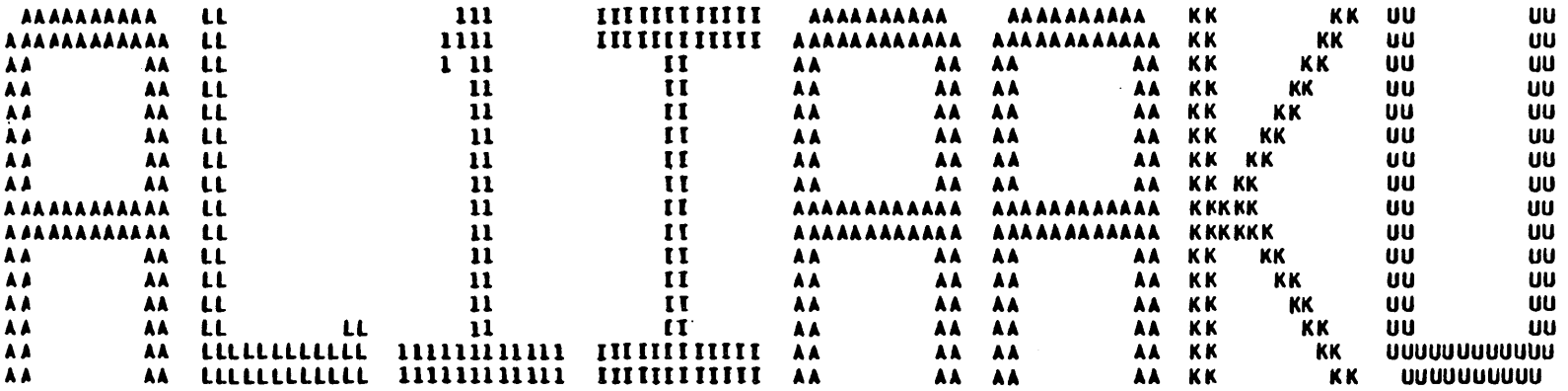
0130 FIRST LINE - FIRST CHANGE

0140 INSERT BETWEEN SECOND AND THIRD LINES

0150

11.47.19.UCLP, 64,023, 0.064KLNS.

```
OPERATING SYSTEM = NOS 4K30-FNT101      81/10/05. PRINTED = 81/10/07. 11.47.20.

UJN          = CCL9      FAMILY      = FEATURE      JOB ORIGIN    = BATCH.
CREATING JSN = AAKS      USER NUMBER = DRF2755      SERVICE CLASS = BATCH.
```



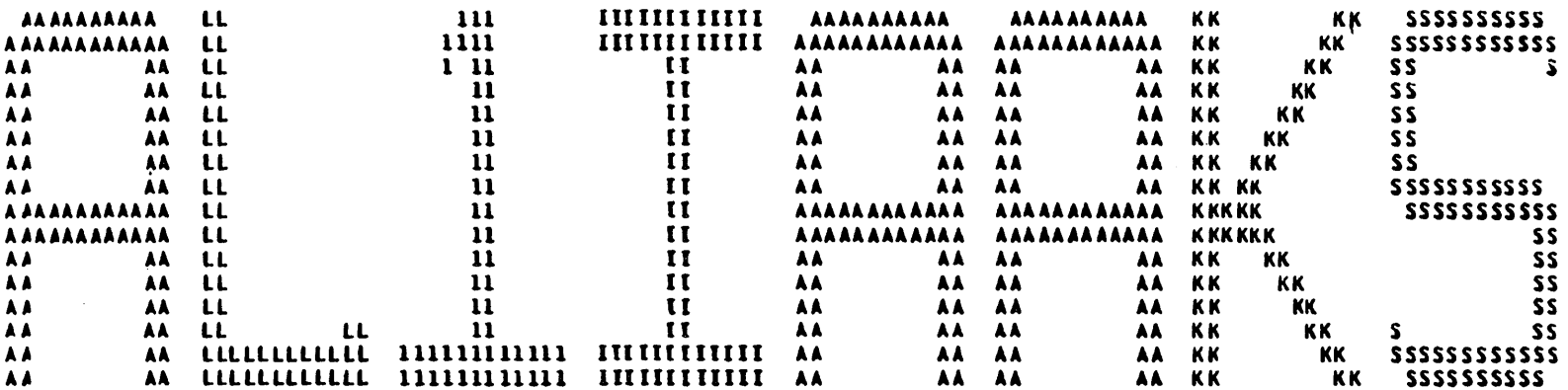

# 2

```
100 FIRST LINE - ORIGIONAL
110 SECOND LINE -ORIGIONAL
120 THIRD LINE - ORIGIONAL
100 FIRST LINE - FIRST CHANGE
115 INSERT BETWEEN SECOND AND THIRD LINES
110
```

```
11.47.22.UCLP, 64.023.      0.064KLNS.
```

OPERATING SYSTEM = NOS 4K30-FNTL)1  81/10/05. PRINTED = 81/10/07. 11.47.24.

UJN = CCL9  FAMILY = FEATURE  JOB ORIGIN = BATCH.
CREATING JSN = AAKS  USER NUMBER = DRF2755  SERVICE CLASS = BATCH.



3

100 FIRST LINE - FIRST CHANGE
115 INSERT BETWEEN SECOND AND THIRD LINES
120 THIRD LINE - ORIGIONAL

# EXERCISE 12

PROBLEM

1. Use C.C.L. statements to iteratively print a file five times.

2. Modify the above jobs' iterative print function to be a callable procedure by this job and which has the number of iterations as a parameter to the procedure.

3. In the example on page 7-22 replace the archaic N.C.L. with C.C.L.

# LESSON 8
# LOADER

## LESSON PREVIEW:

In this lesson the student will learn how programs get loaded into core, how external references are satisfied, how libraries are used, how overlays can be created, and how field length is controlled.

## OBJECTIVES:

After completing this lesson the student should be able to -

- Set up jobs which require more complicated load sequences to satisfy externals.
- Get and use a load map.
- Create and use libraries.
- Create and run an overlay.
- Create and run segments.
- Control his field length using loader control cards.

## TRAINING AIDS:

Visuals V8-1 through V8-25.

## PROJECTS:

Exercise 13 (problem).

## REFERENCES:

CYBER Loader Reference Manual (60429800)

DEFINITION

A LOADER IS A PROGRAM WHICH:

1. BRINGS TOGETHER VARIOUS BLOCKS OF CODE (CALLED RELOCATABLE BINARIES OR COMMON BLOCKS);

2. ORGANIZES THEM ACCORDING TO SOME PRIORITY;

3. CONNECTS THE BLOCKS TOGETHER BY PATCHING IN ADDRESSES TO EXTERNAL REFERENCES.

THE RESULTS OF THIS PROCESS IS ONE CONTINUOUS BLOCK OF CODE CALLED ABSOLUTE BINARY (OR CORE IMAGE).

# EXTERNAL REFERENCE

An External Reference is a reference to a symbol not defined in the program which makes the reference. For FORTRAN, an external reference is constructed whenever the programmer uses a CALL statement. For COBOL users, an external reference is constructed whenever the programmer uses a CALL or ENTER statement. For COMPASS, an external reference is set up whenever the programmer uses an EXT statement or the equivalent. For each external reference, the LOADER will attempt to find a program that defines that symbol as an ENTRY POINT.

EXAMPLE: (FORTRAN)

```
PROGRAM MAIN
+
+
+
CALL SORTALL
+
+
+
+
+
+
STOP
END

SUBROUTINE SORTALL
+
+
+

END
```

The FORTRAN Compiler set up an external reference to SORTALL. The LOADER will look for and find the entry point SORTALL in subroutine SORTALL and will link the two together.

QUESTION: Does the FORTRAN compiler know the address of SORTALL? If it does not, what does it do with the reference to SORTALL?

The FORTRAN compiler set up an Entry Point called SORTALL.

QUESTION: What is an Entry Point?

Routines MAIN and SORTALL are separate logical entities or blocks. They are called PROGRAM BLOCKS or RELOCATABLE BINARIES. After they are assembled, they can be saved as separate binary code entities. Each block of relocatable binary contains a short table at the beginning of the block which contains the name of the routine and a list of one or more entry points to that block. Also included are the addresses of all the instructions in the binary code which make external references. The LOADER can then find these instructions and insert addresses of appropriate entry points. This is called Satisfying Externals.

If any externals other than a weak external remains unsatisfied, a non-fatal error results. Any address field containing an unsatisfied external is filled with addr + $400000_8$, where addr is the address of the reference.

EXAMPLE: SAVING AND USING RELOCATABLE BINARIES ON PERMANENT FILES

```
JOB.
USER(    ,    )
FTN, B=BINMAIN.
SAVE, BINMAIN.
FTN, B=BINSORT.
SAVE, BINSORT.

JOB.
USER(    ,    )
GET, BINMAIN.
GET, BINSORT.
LOAD, BINMAIN, BINSORT.
EXECUTE.
```

EXAMPLE: SAVING AND USING RELOCATABLE BINARIES ON PUNCHED CARDS

```
JOB.
USER.
FTN, B=PUNCHB.
FTN, B=PUNCHB.
7/8/9
   PROGRAM MAIN
   7/8/9
   SUBROUTINE SORTALL
   7/8/9
   6/7/8/9

JOB.
USER.
LOAD (INPUT)
LOAD (INPUT)
EXECUTE.
7/8/9
BINARY FOR PROGRAM MAIN
7/8/9
7/8/9
BINARY FOR SUBROUTINE SORTALL
7/8/9
7/8/9
DATA USED BY PROGRAM MAIN (IF ANY)
6/7/8/9
```

## COMMON BLOCKS

A program or subroutine can also make a reference to data within a Common Block. A COMMON BLOCK is a grouping of data that can be referenced by more than one routine.

COMMON BLOCKS are divided into two types:

BLANK COMMON. This is a block of data that is placed at the very end of a user's field length. The first word of this block is equated with the last word of all executable code, plus one. It cannot be preset with any data. FORTRAN programmers define blank Common with a COMMON A(50) statement.

LABELED COMMON. An example of Labeled Common in FORTRAN is: COMMON/ABC/A(50). Labeled Common Blocks are placed in front of the first routine which defines them. Once defined by one routine, they cannot be redefined to a larger or smaller size by another routine. Data in these blocks can be Preset:

EXAMPLE: DATA A/50*0.0/

COBOL allows a programmer to define one label common block called CCOMMON. He does this by defining a COMMON-STORAGE SECTION.

EXAMPLE: COMMON BLOCKS USED BY FORTRAN AND COBOL

FORTRAN

```
PROGRAM MAIN
+
COMMON/CCOMMON/DATA (3)
+
COMMON/A/A (5)
+
+
CALL SORTALL
C=DATA (1)
+
+
+
STOP
END
```

```
SUBROUTINE SORTALL
COMMON/CCOMMON/DATA (3)
COMMON /A/A (5)
COMMON/SORTALL/SORT (5)
+
+
DATA (1) = 5.0
+
+
RETURN
END
```

COBOL

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  MAIN.
+
+
+
COMMON-STORAGE SECTION.
77 DATA-1 PICTURE IS 9999V99 USAGE IS COMP-1.
77 DATA-2 PICTURE IS 9999V99 USAGE IS COMP-1.
77 DATA-3 PICTURE IS 9999V99 USAGE IS COMP-1.
+
PROCEDURE DIVISION.
INIT SECTION.
INIT-PARA.
+

      ENTER SORTALL.
      MOVE DATA-1 TO C.

+
STOP RUN.
```

## ABSOLUTE BINARY

ABSOLUTE BINARY is a collection of one or more relocatable blocks, organized according to some priority scheme, where references to externals have been "satisfied" wherever possible. EXTERNAL REFERENCES are SATISFIED whenever the loader can find an ENTRY POINT name to match the EXTERNAL REFERENCE name. Otherwise, EXTERNAL REFERENCES are UNSATISFIED.

EXAMPLE: Unsatisfied External

```
JOB
USER
GET,BINMAIN
BINMAIN.
```

When loading BINMAIN, the LOADER will detect that there is an external reference to SORTALL. It will look for SORTALL in BINMAIN and in the system library. Since SORTALL IS NOT IN EITHER place, the external reference remains unsatisfied.

QUESTION: Will the program still go to execution stage even with an unsatisfied external?

QUESTION: If it does go to execution, what will happen when the "CALL SORTALL" statement is executed?

EXAMPLE: Satisfied Externals

```
JOB.
USER.
GET,BINMAIN.
GET,BINSORT.
LOAD (BINSORT)
BINMAIN.
```

QUESTION: Would this sequence work?

```
GET,BINMAIN.          BINMAIN calls SORTALL
GET,BINSORT.          BINSORT contains SORTALL

BINMAIN.
```

## LOAD PRIORITY - LOADER MAP

Blocks are loaded according to the following priority scheme:

1. Labeled Common blocks in the order that they are defined.

2. Program blocks.

3. Blank Common.

A LOADER MAP tells a programmer how the LOADER organized the blocks which were presented to it.

EXAMPLE:

```
PROGRAM MAIN
COMMON/DATA/DATA(5)
COMMON A(6)
+
+
+
END

SUBROUTINE SORTALL
COMMON/DATA/DATA(4)
COMMON B(5),C(2)
COMMON/SORTALL/SORT(5)
+
+
+
RETURN
END
```

LOADER MAP:

| BLOCK: | ADDRESS: | LENGTH: |
|---|---|---|
| /DATA/ | 111 | 5 |
| MAIN | 116 | 100 |
| /SORTALL/ | 216 | 5 |
| SORTALL | 223 | 100 |
| // | 323 | 7 |
| | FWA OF THE LOAD | 111 |
| | LWA OF THE LOAD | 332 |

QUESTIONS:

1. What is a LOADER MAP?

2. How does the LOADER know how long the Blank Common Block is?

3. How does the LOADER know how long Block Data is?

4. What would happen if Subroutine SORTALL defined Block /DATA/ as 10 words?

5. What would happen if Subroutine SORTALL defined the blank Common Block as 10 words?

6. Is it legal for Subroutine SORTALL to identify the data in the Common Block with symbols different than used by Program MAIN?

7. How much field length is required to execute this program?

NOGO (filename, entry point 1, entry point 2 . . .)

NOGO is a command that tells the LOADER to absolutize relocatable binary and put it on one absolute binary file. The programmer can name the absolute file or accept the default name of ABS. The absolute binary can be saved on a permanent file or on binary punched cards.

EXAMPLE: Relocatable and Absolute Binary saved and used

| RELOCATABLE BINARY | ABSOLUTE BINARY |
|---|---|
| JOB.<br>USER( , )<br>FTN. COMPILE MAIN<br>FTN. COMPILE SORTALL<br>SAVE, LGO. | JOB.<br>USER( , )<br>FTN. COMPILE MAIN<br>FTN. COMPILE SORTALL<br>LOAD (LGO)<br>NOGO (BINABS)<br>SAVE, BINABS. |
| JOB.<br>USER( , )<br>GET, LGO.<br>LGO. | JOB.<br>USER( , )<br>GET, BINABS.<br>BINABS. |
| JOB.<br>USER( , )<br>GET, LGO.<br>COPY, LGO, PUNCHB. | JOB.<br>USER( , )<br>GET, BINABS.<br>COPY (BINABS, PUNCHB) |
| JOB.<br>USER( , )<br>INPUT.<br>7/8/9<br>Binary Punched Deck<br>7/8/9<br>7/8/9<br>Data (if any)<br>6/7/8/9 | JOB.<br>USER( , )<br>INPUT.<br>7/8/9<br>Binary Punched Deck<br>7/8/9<br>7/8/9<br>Data (if any)<br>6/7/8/9 |

QUESTIONS:

1. What are the differences between files LGO and BINABS?

2. What are the advantages and disadvantages of saving and using absolute binary versus relocatable binary?

Execute can stand alone as a loader command or can have optional parameters. The entry point parameter gives a programmer the capability of entering relocatable or absolute binary at some entry point other than the default entry point (which is the last encountered transfer address). A transfer address is the address to which control is passed at Execution time. Each FORTRAN and COBOL program is given a transfer address by the compiler. (Transfer addresses are not supplied for subroutines.)

EXAMPLE:

```
JOB.
USER(    ,    )
FTN.
LGO.                Execution begins at program one.
7/8/9
   PROGRAM ONE      QUESTION:  Why?

   SUBROUTINE A

6/7/8/9
```

EXAMPLE:

```
JOB.
USER(    ,    )
FTN.
LGO.                Execution begins at program two.
7/8/9
   PROGRAM ONE      QUESTION:  Why?

   PROGRAM TWO      QUESTION:  Is Program One ever executed in this sequence?

   SUBROUTINE A

6/7/8/9
```

## LOADER SEQUENCE

There are several words reserved for usage by the loader: LOAD, LIBLOAD, SLOAD, CAPSULE, EXECUTE, NOGO, SATISFY, LDSET, SEGLOAD. Once a loader word is used in job control sequence, the remaining job control cards must be loader commands until the loader sequence is terminated by a NOGO or EXECUTE or program name call.

QUESTION: Are the following sequences valid? If not, make them correct.

a.
```
JOB.
USER(    ,    )
GET, BINMAIN.
LOAD (BINMAIN)
GET,BINSORT
LOAD(BINSORT)
EXECUTE.
```

b.
```
JOB.
USER(    ,    )
ATTACH, BINMAIN.
BINMAIN.
EXECUTE.
```

c.
```
JOB.
USER(    ,    )
GET,BINMAIN
GET,BINSORT.
LOAD (BINMAIN)
LOAD (BINSORT)
ABS (ASBMAIN)
ABSMAIN.
```

```
JOB.
USER(    ,    )
ATTACH, BINSORT.
ATTACH, BINMAIN.
BINMAIN.
LOAD (BINMAIN)
EXECUTE.
LOAD (BINSORT)
LOAD (BINMAIN)
EXECUTE (BINMAIN)
```

EXECUTE (entry point, parameter 1, parameter 2 . . .)

It is possible to pass information to a binary program through EXECUTE parameters.

E.G., EXECUTE,ONE,TAPE1,TAPE2.

These parameters are stored in the jobs field length at locations RA+2, RA+3 up through RA+53. In the example, before program one begins to execute, the word TAPE 1 in display code would be placed at RA+2 and TAPE 2 would be placed in RA+3. Both words would be left justified, zero filled with a special code in the lower six bits of the word to indicate what the separator is, i.e., a comma, equal sign, period.

The program can make use of this information. FORTRAN programs make special use of this information. Parameters passed in this fashion redefine files called by the FORTRAN program. The routine Q2NTRY= performs this helpful task.

FORTRAN EXAMPLE:

```
JOB.
USER (    ,    )
GET,TAPE1.
GET,TAPE2.
GET,TAPE3.
GET,FILEA.
GET,FILEB.
FTN.
LOAD(LGO)
EXECUTE,,INPUT,OUTPUT,FILEA.

LGO,INPUT,OUTFILE,FILEB.
REWIND,OUTFILE.
COPY,OUTFILE,OUTPUT.

LOG.
```

QUESTION: What happens at each execution?

7/8/9

```
    PROGRAM ONE(TAPE1,TAPE2,TAPE3)

    READ (1,000).....

    READ (3,3000).....

    WRITE (2,2000).....

    STOP
    END
```

EXAMPLE: Specifying an Entry Point - Relocatable Binary

```
JOB.                                    7/8/9
USER(    ,    )                         PROGRAM ONE
FTN.
LOAD, LGO.                              END
EXECUTE, ONE.                           PROGRAM TWO
LOAD, LGO.
EXECUTE, TWO.
LGO.                                    END
                                        PROGRAM THREE

                                        END
                                        SUBROUTINE A

QUESTION:  Where does EXECUTION         END
           begin with a simple LGO?     7/8/9
                                        DATA FOR PROGRAM ONE
                                        7/8/9
                                        DATA FOR PROGRAM TWO
                                        7/8/9
                                        DATA FOR PROGRAM THREE
                                        6/7/8/9
```

EXAMPLE: Specifying an Entry Point - Absolute Binary

```
JOB . . .
USER . . .
FTN. Same Programs as above Example
LOAD (LGO)
NOGO (ABS, ONE, TWO, THREE)
LOAD (ABS)
EXECUTE, ONE.
LOAD (ABS)
EXECUTE, TWO.
ABS.
```

QUESTION: Where does Execution begin with a simple ABS?

## LOAD - No rewind option

The LOAD command will copy all binaries from a file until an empty record is reached. The default on all loader files is to rewind the file before taking binaries from it. However, a programmer can specify "NR".

EXAMPLE:

```
JOB.
USER (   ,   )
FTN.                  This produces three logical records on the file LGO.
COBOL.

LOAD (LGO)            The entire file is loaded.
EXECUTE.

BKSP (LGO,2)
LOAD (LGO/NR)         Only the COBOL routine is loaded and executed.
EXECUTE.
```

SLOAD (file name, module name 1, module name 2 . . .)

The SLOAD command allows the programmer to selectively load routines from a sequential binary file.

EXAMPLE:

```
JOB.
USER (    ,    )
FTN.                          This will produce six records on LGO.
SLOAD (LGO, ONE, A, B)
EXECUTE.
SLOAD (LGO, THREE, C, B)
EXECUTE.                      QUESTION:  Why use SLOAD rather than LOAD?
7/8/9
     PROGRAM ONE
     PROGRAM TWO
     PROGRAM THREE
     SUBROUTINE A
     SUBROUTINE B
     SUBROUTINE C
6/7/8/9
```

## COPYL (Old Binary File, New Binary File, Updated Binary File)

COPYL is a common utility that is used to update a sequential binary file. It is very helpful in reducing compile time when developing routines that contain several subroutines.

EXAMPLE:

```
JOB.
USER(    ,    )
GET,OLDBIN.                 OLDBIN contains subroutines A, B, C, D. LGO
FTN.                        contains subroutine C. NEWBIN contains A, B, D
COPYL,OLDBIN,LGO,NEWBIN.    from OLDBIN and C from LGO.
NEWBIN.
```

EXAMPLE:

```
JOB.
USER.
GET,OLDPL.                  OLDPL contains program main, subroutines A, B, C,
UPDATE.                     D, Default Update. Only subroutine C was changed
FTN (I)                     and compiled.
GET,OLDBIN.
COPYL,OLDBIN,LGO,NEW.       Only subroutine C is on LGO.
NEW.
REPLACE(NEW=OLDBIN)         Test to see if subroutine C and other routines
                            are still working.
```

LIBLOAD (filename, entry point 1, entry point 2 . . .)

LIBLOAD allows a user to load routines from a library file (as distinguished from a sequential file, where LOAD or SLOAD is used).

EXAMPLE:

```
JOB.
USER(    ,    )
FTN.
GET,LIBFILE.
LIBLOAD(LIBFILE,A,B)
LGO.
```

PROGRAM ONE (which calls A and B) LIBFILE is a library file created by the user on a previous run.

LIBRARY (filename 1, filename 2)

To make loading easier, a user can create a library file and establish that file as part of a GLOBAL LIBRARY SET. This is searched by the loader to satisfy externals. Only two user-created libraries can be placed in the global library set, but additional system libraries can be listed in the set; 24 system libraries and no user libraries or 13 system libraries and 1 user library.

EXAMPLE:

```
JOB.
USER (    ,    )
ATTACH,LIBFIL1.          LIBFIL1 contains routines A, B, C, D, E, F.

ATTACH,LIBFIL2.          LIBFIL2 contains routines G, H, I, J, K, L, M.

LIBRARY(LIBFIL1,LIBFIL2)

FTN.                     Program One (which calls A, F, G, L).

LGO.                     The loader will search LIBFIL1 and LIBFIL2 to
                         satisfy the externals.  A and F will be loaded
                         from LIBFIL1 and G and L will be loaded from
                         LIBFIL2.
```

## LDSET (LIB=library 1 / library 2 / library 3 . . .)

Besides a GLOBAL library set, the Common Loader also recognizes a LOCAL library set. The local library is defined by an LDSET (LIB= ) card. It is defined only for the load sequence.

EXAMPLE:

```
JOB.
USER(   ,   )
ATTACH,LIBSUB1.          Contains routines A, B, C, D, E.
ATTACH,LIBSUB2.          Contains routines A, B, C, G, H, I.
FTN.                     Program calls A and B.
                         Routines A, B, C are slightly different on each
                         library.

LDSET(LIB=LIBSUB1)       Only LIBSUB1 is searched.
LGO.

LDSET(LIB=LIBSUB2)       Only LIBSUB2 is searched.
LGO.

LGO.                     External references to A and B are left unsatisfied.

6/7/8/9
```

LDSET (SUBST = ORIGINAL REFERENCE - REPLACEMENT/ . . .)

EXAMPLE:

```
JOB.
USER (    ,    )
ATTACH, NEWSORT.                  Contains subroutine NEWSORT.
ATTACH, BINMAIN.                  Binmain calls SORTALL.
LDSET(SUBST=SORTALL - NEWSORT)
LOAD (NEWSORT)
BINMAIN.
6/7/8/9
```

## SATISFY (library 1, library 2, . . .)

Satisfy causes External references to be satisfied from specified library files for all routines loaded up to that point.

EXAMPLE:

```
JOB.
USER (    ,    )
ATTACH, BINONE.      Calls routines A and B.
ATTACH, BINTWO.      Calls routines C and D.
ATTACH, LIBFIL1.     Contains routines A, B, C, D.
ATTACH, LIBFIL2.     Contains slightly different routines A, B, C, D.

LOAD (BINONE)
SATISFY (LIBFIL2)    QUESTION: From which libraries are routines A, B, C, D
                     loaded?
LOAD (BINTWO)
SATISFY (LIBFIL1)
EXECUTE.
6/7/8/9
```

```
LDSET (OMIT=Ext. Ref 1/Ext. Ref. 2/ . . .)
LDSET (USEP=routine 1/routine 2/ . . .)
LDSET (USE=Entry point 1/ Entry point 2/ . . .)
```

These options give the programmer more powerful tools for selection routines from libraries.

EXAMPLE:

```
JOB.
USER (    ,    )
ATTACH, LIB1.              LIB1 contains A, B, C, D, E.
ATTACH, LIB2.              LIB2 contains A, G, H, I, J, K, C, D.

ATTACH, BIN.               BIN calls A, B, C, C, E, G, H, I, J.

LIBRARY (LIB1, LIB2)       We want all routines from LIB1 except A.

LDSET (OMIT=A)             A and other routines not yet satisfied can come
                           from LIB2.
LOAD (BIN)
SATISFY.
LDSET (USE=A) or LDSET (USEP=A)
SATISFY (LIB2)
EXECUTE.
6/7/8/9
```

## EXTERNALS WHICH CALL EXTERNALS

Suppose a programmer has three libraries:

LIB1 - Contains A, B, C, D, E, ONE, TWO, JANE, BETTY

LIB2 - Contains ONE, TWO, THREE, FOUR, FIVE, JANE, BETTY

LIB3 - Contains JANE, BETTY, MARY, A, B

Suppose:

A calls JANE, FOUR and MARY

ONE calls TWO, E and B

MARY calls B, D and ONE

Assume the following card sequence:

```
JOB.
USER (    ,    )
ATTACH, LIB1.
ATTACH, LIB2.
ATTACH, LIB3.
FTN.
LDSET (LIB=LIB1/LIB2/LIB3)
LGO.
7/8/9
    PROGRAM MAIN (OUTPUT)

    CALL A

    END
6/7/8/9
```

QUESTION: From which libraries are the above routines satisfied?

## GLOBAL versus LOCAL Libraries

GLOBAL library sets are defined by a LIBRARY card. The set is limited to two users and two system libraries, one user and 13 system or no user and 24 system libraries. No limit exists for a local library set. It is defined by LDSET (LIB= ).

The GLOBAL library set remains in effect until redefined by another LIBRARY card. To empty the Global Library Set use LIBRARY. LIBRARY may not appear in a load sequence.

EXAMPLE:

```
JOB.
USER (    ,    )
ATTACH,LIB1.
ATTACH,LIB2.
LIBRARY(LIB2)                 The Global Library = LIB 2

LIBRARY(LIB1, LIB2)           The Global Library = LIB 1, LIB 2

LIBRARY.                      The Global Library is empty.
```

The Local Library Set remains in effect only until the end of the Load Sequence and affects only those routines loaded after the definition of the library. Additional LDSET(LIB= ) cards within a load sequence add to the LOCAL library set; they do not redefine it.

EXAMPLE:

```
JOB.
USER(    ,    )
ATTACH, LIB1.
ATTACH, LIB2.
GET(LGO)
LDSET(LIB=LIB1) Local Library = LIB1
LOAD(LGO)
LOAD(BIN)
EXECUTE.

LOAD(LGO)                     No Local Library Set; LIB1 will not be searched
LDSET(LIB=LIB1)               for Externals from LGO.

LOAD(BINA)                    The Loader will search LIB1 for Externals from
                              BINA.

LDSET(LIB=LIB2)               Local Library = LIB1, LIB2

LOAD(BINB)                    The Loader will search LIB1, LIB2 for Externals
                              from BINB.
```

## SATISFYING EXTERNALS - PRIORITY

1. Local file, i.e., the binaries loaded through direct load calls:

   EXAMPLE:

```
LOAD (BINONE)              These binaries are the LOCAL FILE.
LOAD (BINTWO)
```

2. GLOBAL Library Set (defined by LIBRARY (library 1, . . .)).

3. Local Library Set (defined by LDSET (LIB=library 1, . . .)).

4. Default Library Set

   EXAMPLE:

```
JOB.
USER (    ,    )
ATTACH, LIBONE.          Contains routines A, B, C, D, E, F, G.

ATTACH, LIBTWO.          Contains routines A, B, C, D, E, F, G.

ATTACH, LIBTHR.          Contains routines A, B, C, D, E, F, G.

ATTACH, LIBFOUR.         Contains routines A, B, C, D, E, F, G.

ATTACH, BIN A.           Calls A.
ATTACH, BIN B.           Calls B.
ATTACH, BIN C.           Calls C.
ATTACH, BIN D.           Calls D.
LDSET (LIB=LIBONE)
BIN A.

LIBRARY (LIBTWO)
LDSET (LIB=LIBONE)
BIN B.

LIBRARY (LIBTHR, LIBFOUR)
BIN C.

LIBRARY.
LOAD (BINA)
SATISFY (LIBFOUR)
LOAD (BINB)
LIBLOAD (LIBTWO,B)
LDSET (LIB=LIBONE)
LOAD ( BINC)
LOAD ( BIND)
EXECUTE.
```

   QUESTION: How are Externals satisfied during each load sequence?

```
LOAD3,T5.
USER,DRF2755,DRF.
MAP(PART)
GET(A=USELIB1,B=USELIB2,C=USELIB3)
LIBLOAD(A,MAIN,SUB1,SUB2,SUB3,SUB4)
EXECUTE,MAIN.
LIBLOAD(B,YOUMAIN,SUB1A,SUB2A,SUB3A,SUB4A)
EXECUTE.
LIBRARY(C)
LIBLOAD(C,OURMAIN)
EXECUTE(OURMAIN)
Q     6/7/8/9 END-OF-INFORMATION
```

```
      FULL  DAYFILE.  81/10/07. 11.44.34.*11.44.27* PAGE     1

 11.44.27.LOAD3,T5.
 11.44.27.UCCR, 64020,       0.011KCDS.
 11.44.27.USER,DRF2755,.
 11.44.28.MAP(PART)
 11.44.28.GET(A=USELIB1,B=USELIB2,C=USELIB3)
 11.44.28.LIBLOAD(A,MAIN,SUB1,SUB2,SUB3,SUB4)
 11.44.28.EXECUTE,MAIN.
 11.44.30.      END   MAIN
 11.44.30.    014300 MAXIMUM EXECUTION FL.
 11.44.30.      0.033 CP SECONDS EXECUTION TIME.
 11.44.30.LIBLOAD(B,YOUMAIN,SUB1A,SUB2A,SUB3A,SUB4A)
 11.44.30.EXECUTE.
 11.44.32.      END   YOUMAIN
 11.44.32.    014300 MAXIMUM EXECUTION FL.
 11.44.32.      0.035 CP SECONDS EXECUTION TIME.
 11.44.32.LIBRARY(C)
 11.44.32.LIBLOAD(C,OURMAIN)
 11.44.32.EXECUTE(OURMAIN)
 11.44.33.      END   OURMAIN
 11.44.33.    014300 MAXIMUM EXECUTION FL.
 11.44.33.      0.040 CP SECONDS EXECUTION TIME.
 11.44.34.UEAD,        0.001KUNS.
 11.44.34.UEPF,        0.027KUNS.
 11.44.34.UEMS,        2.290KUNS.
 11.44.34.UECP,        1.399SECS.
 11.44.34.AESR,        2.976UNTS.
 11.44.34.$OUT(*/OP=E)
 11.44.34. NO FILES PROCESSED.
 11.44.34.$DAYFILE(OUTPUT,JT=D)
 11.45.49.UCLP, 64,023,       0.704KLNS.
```

1

2

3

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 10162

1

TRANSFER ADDRESS -- MAIN 2174

PROGRAM ENTRY POINTS -- MAIN 2174

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| MAIN | 111 | 2101 | UL-A | 81/10/07 | FTN | 4.8 552 | 666X I | PROGRAM OPT=1 |
| SUB1 | 2212 | 14 | UL-A | 81/10/07 | FTN | 4.8 552 | 666X I | SUBROUTINEOPT=1 |
| SUB2 | 2226 | 14 | UL-A | 81/10/07 | FTN | 4.8 552 | 666X I | SUBROUTINEOPT=1 |
| SUB3 | 2242 | 14 | UL-A | 81/10/07 | FTN | 4.8 552 | 666X I | SUBROUTINEOPT=1 |
| SUB4 | 2256 | 14 | UL-A | 81/10/07 | FTN | 4.8 552 | 666X I | SUBROUTINEOPT=1 |
| SYSAID= | 2272 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2273 | 1 | | | | | | |
| /FCL.C./ | 2274 | 30 | | | | | | |
| /Q8.IO./ | 2324 | 130 | | | | | | |
| Q2NTRY= | 2454 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 552 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2454 | 43 | | | | | | |
| COMIO= | 2517 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2552 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FCL CAPSULE LOADING |
| FECMSK= | 2612 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2653 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2656 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON FLOATING OUTPUT CODE |
| FMTAP= | 3167 | 357 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | 3546 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 4247 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FCL MISC. UTILITIES. |
| GETFIT= | 4274 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | 4370 | 470 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | OUTPUT FORMAT INTERPRETER. |
| LDOUT= | 5060 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 5324 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON OUTPUT CODE |
| OUTC= | 5500 | 162 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FORMATTED WRITE FORTRAN RECORD. |
| OUTF= | 5662 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 6066 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 6073 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 6133 | 14 | | | | | | |
| FDL.RES | 6147 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 6360 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 6602 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | 7237 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| ERR$RM | 7316 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 7343 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 7432 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 7437 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 7621 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 7627 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 7643 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 7701 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 7723 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | |
| CMM.R | 7732 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 10140 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

.411 CP SECONDS 260008 CM STORAGE USED 4 TABLE MOVES

```
IN MAIN
 IN SUB1-PROGRAM MAIN
 IN SUB2-PROGRAM MAIN
 IN SUB3-PROGRAM MAIN
 IN SUB4-PROGRAM MAIN
```

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 10162

TRANSFER ADDRESS -- YOUMAIN 2174

PROGRAM ENTRY POINTS -- YOUMAIN 2174

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| YOUMAIN | 111 | 2101 | UL-B | 81/10/07 | FTN | 4.8 | 552 | 666X I | PROGRAM OPT=1 |
| SUB1A | 2212 | 14 | UL-B | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB2A | 2226 | 14 | UL-B | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB3A | 2242 | 14 | UL-B | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB4A | 2256 | 14 | UL-B | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SYSAID= | 2272 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2273 | 1 | | | | | | | |
| /FCL.C./ | 2274 | 30 | | | | | | | |
| /Q8.ID./ | 2324 | 130 | | | | | | | |
| Q2NTRY= | 2454 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2454 | 43 | | | | | | | |
| COMIO= | 2517 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2552 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL CAPSULE LOADING |
| FECMSK= | 2612 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2653 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2656 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON FLOATING OUTPUT CODE |
| FMTAP= | 3167 | 357 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | 3546 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 4247 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL MISC. UTILITIES. |
| GETFIT= | 4274 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | 4370 | 470 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | OUTPUT FORMAT INTERPRETER. |
| LDOUT= | 5060 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 5324 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON OUTPUT CODE |
| OUTC= | 5500 | 162 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORMATTED WRITE FORTRAN RECORD. |
| OUTF= | 5662 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 6066 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PRICE |
| CPU.SYS | 6073 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 6133 | 14 | | | | | | | |
| FDL.RES | 6147 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 6360 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 6602 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | 7237 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| ERR$RM | 7316 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 7343 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 7432 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 7437 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 7621 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 7627 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 7643 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 7701 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 7723 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | 7732 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 10140 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |

.404 CP SECONDS 26000B C4 STORAGE USED 4 TABLE MOVES

```
IN YOUMAIN
 IN SUB1A-PROGRAM YOUMAIN
 IN SUB2A-PROGRAM YOUMAIN
 IN SUB3A-PROGRAM YOUMAIN
 IN SUB4A-PROGRAM YOUMAIN
```

3

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 10177

TRANSFER ADDRESS -- OURMAIN 2174

PROGRAM ENTRY POINTS -- OURMAIN 2174

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| OURMAIN | 111 | 2102 | UL-C | 81/10/07 | FTN | 4.8 | 552 | 666X I | PROGRAM OPT=1 |
| SUB3B | 2213 | 14 | UL-C | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB2B | 2227 | 14 | UL-C | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB3B | 2243 | 14 | UL-C | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB4B | 2257 | 14 | UL-C | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB5B | 2273 | 14 | UL-C | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SYSAID= | 2307 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 2310 | 1 | | | | | | | |
| /FCL.C./ | 2311 | 30 | | | | | | | |
| /Q8.IO./ | 2341 | 130 | | | | | | | |
| Q2NTRY= | 2471 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2471 | 43 | | | | | | | |
| COMIO= | 2534 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 2567 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL CAPSULE LOADING |
| FECMSK= | 2627 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | 2670 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CONVERTED DATA STORAGE |
| FLTOUT= | 2673 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON FLOATING OUTPUT CODE |
| FMTAP= | 3204 | 357 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | 3563 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 4264 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL MISC. UTILITIES. |
| GETFIT= | 4311 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | 4405 | 470 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | OUTPUT FORMAT INTERPRETER. |
| LDOUT= | 5075 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 5341 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON OUTPUT CODE |
| OUTC= | 5515 | 162 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORMATTED WRITE FORTRAN RECORD. |
| OUTF= | 5677 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPM | 6103 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| CPU.SYS | 6110 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 6150 | 14 | | | | | | | |
| FDL.RES | 6164 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 6375 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 6617 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | 7254 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| ERR$RM | 7333 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 7360 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 7447 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 7454 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 7636 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 7644 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 7660 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 7716 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | 7740 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | 7747 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|
| CMF.SLF | 10155 | 22 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMN V1.1 - SHRINK AT LWA FIXED. |

.429 CP SECONDS 260008 CM STORAGE USED 6 TABLE MOVES

```
IN OURMAIN
 IN SUB1B-PROGRAM OURMAIN
 IN SUB2B-PROGRAM OURMAIN
 IN SUB3B-PROGRAM OURMAIN
 IN SUB4B-PROGRAM OURMAIN
 IN SUB5B-PROGRAM OURMAIN
```

```
LOAD4,T5.
USER,DRF2755,DRF.
GET(USELIB1,USELIB2,USELIB3)
MAP(PART)
LDSET(LIB=USELIB1/USELIB2/USELIB3)
LDSET(SUBST=SUB1B-SUB1/SUB2B-SUB2A/SUB3B-SUB3/SUB4B-SUB4A)
LIBLOAD(USELIB3,OURMAIN)
NOGO(LOG)
REWIND(LOG)
LOAD(LOG)
EXECUTE(OURMAIN)
LOG.
Q     6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/07. 11.45.14.*11.45.09* PAGE    1

11.45.09.LOAD4,T5.
11.45.09.UCCR, 64020,       0.012KCDS.
11.45.09.USER,DRF2755,.
11.45.09.GET(USELIB1,USELIB2,USELIB3)
11.45.10.MAP(PART)
11.45.10.LDSET(LIB=USELIB1/USELIB2/USELIB3)
11.45.10.LDSET(SUBST=SUB1B-SUB1/SUB2B-SUB2A/SUB3B-SUB3/SUB4B-SUB4A)  1
11.45.10.LIBLOAD(USELIB3,OURMAIN)
11.45.10.NOGO(LOG)
11.45.12.REWIND(LOG)
11.45.12.LOAD(LOG)
11.45.12.EXECUTE(OURMAIN)
11.45.13.      END  OURMAIN  2
11.45.13.     014300 MAXIMUM EXECUTION FL.
11.45.13.      0.032 CP SECONDS EXECUTION TIME.
11.45.13.LOG.
11.45.14.      END  OURMAIN  3
11.45.14.     014300 MAXIMUM EXECUTION FL.
11.45.14.      0.037 CP SECONDS EXECUTION TIME.
11.45.14.UEAD,       0.001KUNS.
11.45.14.UEPF,       0.027KUNS.
11.45.14.UEMS,       2.116KUNS.
11.45.14.UECP,       0.745SECS.
11.45.14.AESR,       2.230UNTS.
11.45.14.$OUT(*/OP=E)
11.45.14. NO FILES PROCESSED.
11.45.14.$DAYFILE(OUTPUT,JT=D)
11.46.08.UCLP, 64,023,      0.384KLNS.
```

1

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 10177

WRITTEN TO FILE LOG

TRANSFER ADDRESS -- OURMAIN 2174

PROGRAM ENTRY POINTS -- OURMAIN 2174

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| OURMAIN | 111 | 2102 | UL-USELIB3 | 81/10/07 | FTN | 4.8 | 552 | 666X I | PROGRAM OPT=1 |
| SUB1 | 2213 | 14 | UL-USELIB1 | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB3 | 2227 | 14 | UL-USELIB1 | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB2A | 2243 | 14 | UL-USELIB2 | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB4A | 2257 | 14 | UL-USELIB2 | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SUB5B | 2273 | 14 | UL-USELIB3 | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| /STP.END/ | 2307 | 1 | | | | | | | |
| /FCL.C./ | 2310 | 30 | | | | | | | |
| /Q8.IO./ | 2340 | 130 | | | | | | | |
| Q2NTRY= | 2470 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 2470 | 43 | | | | | | | |
| FORSYS= | 2533 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| OUTC= | 3234 | 162 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORMATTED WRITE FORTRAN RECORD. |
| OUTF= | 3416 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT CONTROL |
| SYSAID= | 3622 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| COMIO= | 3623 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FDL | 3656 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL CAPSULE LOADING |
| FEIFST= | 3716 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CONVERTED DATA STORAGE |
| FMTAP= | 3721 | 357 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORUTL= | 4300 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL MISC. UTILITIES. |
| GETFIT= | 4325 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | 4421 | 470 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | OUTPUT FORMAT INTERPRETER. |
| LDOUT= | 5111 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 5355 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON OUTPUT CODE |
| FECMSK= | 5531 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | INITIALIZE CONSTANTS. |
| FLTOUT= | 5572 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON FLOATING OUTPUT CODE |
| CPU.SYS | 6103 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 6143 | 14 | | | | | | | |
| FDL.MMI | 6157 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | 6401 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | 7036 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| CMF.GSS | 7115 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CPU.CPM | 7137 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PRICE |
| FDL.RES | 7144 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| ERR$RM | 7355 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 7402 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 7471 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 7476 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | 7660 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMF.FRF | 7666 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMM.R | 7724 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| CMF.SLF | 10132 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |
| CMM.FFA | 10154 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMM.MEM | 10170 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 528 | | |

.581 CP SECONDS 257008 CM STORAGE USED 4 TABLE MOVES

2

```
IN OURMAIN
 IN SUB1-PROGRAM MAIN
 IN SUB2A-PROGRAM YOUMAIN
 IN SUB3-PROGRAM MAIN
 IN SUB4A-PROGRAM YOUMAIN
 IN SUB5B-PROGRAM OURMAIN
```

3

```
IN OURMAIN
 IN SUB1-PROGRAM MAIN
 IN SUB2A-PROGRAM YOUMAIN
 IN SUB3-PROGRAM MAIN
 IN SUB4A-PROGRAM YOUMAIN
 IN SUB5B-PROGRAM OURMAIN
```

```
OVRLY1,T5.
USER,ORF2755,ORF.
MAP(PART)
FTN,R=0.
LOAD(LGO)
NOGO.
DEMO.
SAVE(DEMO)
EXIT.
REWIND(DEMO)
DEFINE(DEMOPF)
COPY(DEMO,DEMOPF,VERIFY)
R     7/8/9 END-OF-RECORD
      OVERLAY (DEMO,0,0)
      PROGRAM MAIN(OUTPUT,INPUT,TAPE1=INPUT)
      COMMON /BLK1/ III(5)
      COMMON /BLK2/ JJJ(5)
      COMMON // IRESULT(2)
      PRINT 100
  100 FORMAT(1H1)
      CALL ONE(1)
      READ (1,*) (III(I),I=1,5)
      CALL ONE(2)
      CALL OVERLAY (4HDEMO,1,0)
      READ (1,*) (JJJ(I),I=1,5)
      CALL ONE(3)
      CALL OVERLAY (4HDEMO,2,0)
      ITOTAL = IRESULT(1) + IRESULT(2)
      PRINT 50,ITOTAL
   50 FORMAT(*  ITOTAL= *,I15)
      CALL ONE(4)
      PRINT *,(III(I),I=1,5),(JJJ(I),I=1,5)
      END
      SUBROUTINE ONE(K)
      GO TO (10,20,30,40,50,60,70,80,90) ,K
   10 PRINT *," ENTERED MAIN"
      GO TO 99
   20 PRINT *," CALL PRIM10"
      GO TO 99
   30 PRINT *," CALL PRIM20"
      GO TO 99
   40 PRINT *," THE END"
      GO TO 99
   50 PRINT *," ENTERED PRIM10"
      GO TO 99
   60 PRINT *," ENTERED SEC11"
      GO TO 99
   70 PRINT *," ENTERED SEC12"
      GO TO 99
   80 PRINT *," ENTERED PRIM20"
      GO TO 99
   90 PRINT *," ENTERED SEC21"
   99 RETURN
      END
      OVERLAY (DEMO,1,0)
      PROGRAM PRIM10
      COMMON /BLK1/ II(5)
      COMMON // IRES(2)
      CALL ONE(5)
      IRES(1)=0
      DO 10 I=1,5
   10 IRES(1)=IRES(1)+II(I)
      CALL OVERLAY(4HDEMO,1,1)
      CALL OVERLAY(4HDEMO,1,2)
      END
      OVERLAY (DEMO,1,1)
      PROGRAM SEC11
      CALL ONE(6)
      END
      OVERLAY (DEMO,1,2)
      PROGRAM SEC12
      CALL ONE(7)
      END
      OVERLAY (DEMO,2,0)
      PROGRAM PRIM20
      COMMON /BLK2/ JJ(5)
      COMMON // JRES(2)
      CALL ONE(8)
      JRES(2)=0
      DO 10 J=1,5
   10 JRES(2)=JRES(2) + JJ(J)
      CALL OVERLAY (4HDEMO,2,1)
      END
      OVERLAY (DEMO,2,1)
      PROGRAM SEC21
      CALL ONE(9)
      END
R     7/8/9 END-OF-RECORD
   10  20  30  40  50
   60  70  80  90 100
0     6/7/8/9 END-OF-INFORMATION
```

```
          FULL  DAYFILE.  81/10/07. 11.16.10.*11.15.51* PAGE    1

 11.15.51.OVRLY1,T5.
 11.15.51.UCCR, 72070,       0.088KCDS.
 11.15.52.USER,ORF2755,.
 11.15.52.MAP(PART)
 11.15.52.FTN,R=0.
 11.16.01.       .712 CP SECONDS COMPILATION TIME
 11.16.01.LOAD(LGO)
 11.16.01.NOGO.
 11.16.06.DEMO.
 11.16.08.      END  MAIN
 11.16.08.     022600 MAXIMUM EXECUTION FL.
 11.16.08.073740.96R CP SECONDS EXECUTION TIME.
 11.16.08.SAVE(DEMO)
 11.16.09.EXIT.
 11.16.09.UEAD,       0.001KUNS.
 11.16.09.UEPF,       0.045KUNS.
 11.16.09.UEMS,       6.506KUNS.
 11.16.09.UECP,       2.128SECS.
 11.16.09.AESR,       4.758UNTS.
 11.16.09.$CUT(*/OP=E)
 11.16.10. NO FILES PROCESSED.
 11.16.10.$DAYFILE(OUTPUT,JT=D)
 11.16.33.UCLF, 72,073,       0.896KLNS.
```

1

2

```
     PROGRAM MAIN       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.15.52       PAGE     1

 1                OVERLAY (DEMO,0,0)
                  PROGRAM MAIN(OUTPUT,INPUT,TAPE1=INPUT)
                  COMMON /BLK1/ III(5)
                  COMMON /BLK2/ JJJ(5)
 5                COMMON // IRESULT(2)
                  PRINT 100
                100 FORMAT(1H1)
                  CALL ONE(1)
                  READ (1,*) (III(I),I=1,5)
10                CALL ONE(2)
                  CALL OVERLAY (4HDEMO,1,0)
                  READ (1,*) (JJJ(I),I=1,5)
                  CALL ONE(3)
                  CALL OVERLAY (4HDEMO,2,0)
15                ITOTAL = IRESULT(1) + IRESULT(2)
                  PRINT 50,ITOTAL
                 50 FORMAT(*  ITOTAL= *,I15)
                  CALL ONE(4)
                  PRINT *,(III(I),I=1,5),(JJJ(I),I=1,5)
20                END
```

1



```
  SUBROUTINE ONE       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.15.52       PAGE     1

 1                SUBROUTINE ONE(K)
                  GO TO (10,20,30,40,50,60,70,80,90) ,K
                 10 PRINT *,≠ ENTERED MAIN≠
                  GO TO 99
 5               20 PRINT *,≠ CALL PRIM10≠
                  GO TO 99
                 30 PRINT *,≠ CALL PRIM20≠
                  GO TO 99
                 40 PRINT *,≠ THE END≠
10                GO TO 99
                 50 PRINT *,≠ ENTERED PRIM10≠
                  GO TO 99
                 60 PRINT *,≠ ENTERED SEC11≠
                  GO TO 99
15               70 PRINT *,≠ ENTERED SEC12≠
                  GO TO 99
                 80 PRINT *,≠ ENTERED PRIM20≠
                  GO TO 99
                 90 PRINT *,≠ ENTERED SEC21≠
20               99 RETURN
                  END
```

```
      PROGRAM PRIM10      73/74   OPT=1

 1              OVERLAY (DEMO,1,0)
                PROGRAM PRIM10
                COMMON /BLK1/ II(5)
                COMMON // IRES(2)
 5              CALL ONE(5)
                IRES(1)=0
                DO 10 I=1,5
             10 IRES(1)=IRES(1)+II(I)
                CALL OVERLAY(4HDEMO,1,1)
10              CALL OVERLAY(4HDEMO,1,2)
                END
```

FTN 4.8+552 81/10/07. 11.15.52 PAGE 1

```
      PROGRAM SEC11       73/74   OPT=1

 1              OVERLAY (DEMO,1,1)
                PROGRAM SEC11
                CALL ONE(6)
                END
```

FTN 4.8+552 81/10/07. 11.15.52 PAGE 1

```
      PROGRAM SEC12       73/74   OPT=1

 1              OVERLAY (DEMO,1,2)
                PROGRAM SEC12
                CALL ONE(7)
                END
```

FTN 4.8+552 81/10/07. 11.15.52 PAGE 1

```
      PROGRAM PRIM20      73/74   OPT=1

 1              OVERLAY (DEMO,2,0)
                PROGRAM PRIM20
                COMMON /BLK2/ JJ(5)
                COMMON // JRES(2)
 5              CALL ONE(8)
                JRES(2)=0
                DO 10 J=1,5
             10 JRES(2)=JRES(2) + JJ(J)
                CALL OVERLAY (4HDEMO,2,1)
10              END
```

FTN 4.8+552 81/10/07. 11.15.52 PAGE 1

```
      PROGRAM SEC21       73/74   OPT=1

 1              OVERLAY (DEMO,2,1)
                PROGRAM SEC21
                CALL ONE(9)
                END
```

FTN 4.8+552

-------- OVERLAY(DEMO,0,0)

FWA OF THE LOAD  111
LWA+1 OF THE LOAD  14735

TRANSFER ADDRESS -- MAIN  4264

PROGRAM ENTRY POINTS --  MAIN  4264

PROGRAM AND BLOCK ASSIGNMENTS:

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| /BLK1/ | 111 | 5 | | | | | | | |
| /BLK2/ | 116 | 5 | | | | | | | |
| MAIN | 123 | 4256 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | PROGRAM OPT=1 |
| ONE | 4401 | 143 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SYSAID= | 4544 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| /STP.END/ | 4545 | 1 | | | | | | | |
| /FCL.C./ | 4546 | 30 | | | | | | | |
| /Q8.IO./ | 4576 | 130 | | | | | | | |
| Q2NTRY= | 4726 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | FCL INITIALIZATION ROUTINE. |
| /FCL=ENT/ | 4726 | 43 | | | | | | | |
| COMIC= | 4771 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FCL | 5024 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL CAPSULE LOADING |
| FECMSK= | 5064 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | 5125 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CONVERTED DATA STORAGE |
| FLTIN= | 5130 | 156 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON FLOATING INPUT CONVERTER. |
| FLTOUT= | 5306 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON FLOATING OUTPUT CODE |
| FMTAF= | 5617 | 357 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | 6176 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | 6677 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL MISC. UTILITIES. |
| GETFIT= | 6724 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| GOTCER= | 7020 | 15 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMPUTED GO TO ERROR PROCESSOR. |
| INCOM= | 7035 | 145 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON INPUT FORMATTING CODE |
| INPF= | 7202 | 250 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED INPUT CONTROL |
| KODER= | 7452 | 470 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | OUTPUT FORMAT INTERPRETER. |
| LOIN= | 10142 | 260 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED INPUT FORMATTING |
| LOOUT= | 10422 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | 10666 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON OUTPUT CODE |
| OUTC= | 11042 | 162 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORMATTED WRITE FORTRAN RECORD. |
| OUTF= | 11224 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT CONTROL |
| OVERLAY | 11430 | 260 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | OVERLAY LOADING ROUTINE. |
| CPU.CPM | 11710 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| CPU.MVE | 11715 | 64 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | MOVE BLOCK OF DATA. |
| CPU.SYS | 12001 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| /FDL.COM/ | 12041 | 14 | | | | | | | |
| FDL.RES | 12055 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | 12266 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| FOL.RES | 12510 | 124 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST OVERLAY LOADER RESIDENT. |
| CTL$RM | 12634 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | 13271 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| CTL$WR | 13350 | 44 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - WEOX, REWIND |

OVERLAY(DEMO,0,0)

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|
| ERR$RM | 13414 | 25 | SL-SYSLIB | 80/09/25 | COMPASS 3.6 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | 13441 | 67 | SL-SYSLIB | 80/09/25 | COMPASS 3.6 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | 13530 | 5 | SL-SYSLIB | 80/09/25 | COMPASS 3.6 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | 13535 | 162 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMM.CIA | 13717 | 105 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - CHANGE INTERNAL AREA. |
| CMF.CSF | 14024 | 6 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | 14032 | 14 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | 14046 | 36 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | 14104 | 22 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMF.LDV | 14126 | 271 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - LOAD OVERLAY. |
| CMF.LCV | 14417 | 55 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - LOAD OVERLAY VIA FOL. |
| CMM.MEM | 14474 | 7 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | |
| CMM.R | 14503 | 206 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | 14711 | 22 | SL-SYSLIB | 80/11/25 | COMPASS 3.6 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |
| // | 14733 | 2 | | | | | |

-------- OVERLAY(DEMO,1,0)

FWA OF THE LOAD 14742
LWA+1 OF THE LOAD 15011

TRANSFER ADDRESS -- PRIM10 14750

PROGRAM AND BLOCK ASSIGNMENTS:

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|
| PRIM10 | 14742 | 47 | LGO | 81/10/07 | FTN 4.8 552 | 666X I | PROGRAM OPT=1 |

-------- OVERLAY(DEMO,1,1)

FWA OF THE LOAD 15016
LWA+1 OF THE LOAD 15033

TRANSFER ADDRESS -- SEC11 15024

PROGRAM AND BLOCK ASSIGNMENTS:

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|
| SEC11 | 15016 | 15 | LGO | 81/10/07 | FTN 4.8 552 | 666X I | PROGRAM OPT=1 |

```
OVERLAY(DEMO,1,2)

-------- OVERLAY(DEMO,1,2)

         FWA OF THE LOAD          15016
         LWA+1 OF THE LOAD        15033

         TRANSFER ADDRESS -- SEC12          15024


         PROGRAM AND BLOCK ASSIGNMENTS.

         BLOCK       ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS
         SEC12         15016       15    LGO       81/10/07 FTN      4.8 552    666X I     PROGRAM   OPT=1


-------- OVERLAY(DEMO,2,0)

         FWA OF THE LOAD          14742
         LWA+1 OF THE LOAD        15001

         TRANSFER ADDRESS -- PRIM20         14750


         PROGRAM AND BLOCK ASSIGNMENTS.

         BLOCK       ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS
         PRIM20        14742       37    LGO       81/10/07 FTN      4.8 552    666X I     PROGRAM   OPT=1


-------- OVERLAY(DEMO,2,1)

         FWA OF THE LOAD          15006
         LWA+1 OF THE LOAD        15023

         TRANSFER ADDRESS -- SEC21          15014


         PROGRAM AND BLOCK ASSIGNMENTS.
```

```
LOAD MAP - MAIN                                CYBER LOADER 1.5-552      81/10/07. 11.16.04.      PAGE     4
OVERLAY(DEMO,2,1)

BLOCK       ADDRESS    LENGTH     FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS
SEC21         15006        15     LGO       81/10/07 FTN      4.8 552    666X I     PROGRAM   OPT=1

     1.198 CP SECONDS              36100B CM STORAGE USED                    19 TABLE MOVES


 ENTERED MAIN
 CALL PRIM10
 ENTERED PRIM10
 ENTERED SEC11
 ENTERED SEC12
 CALL PRIM20
 ENTERED PRIM20
 ENTERED SEC21
 ITOTAL=              550
 THE END
10 20 30 40 50 60 70 80 90 100
```

```
OVRLY2,T5.
USER,DRF2755,DRF.
MAP(PART)
GET,DEMO.
DEMO.
DMP(200)              ** DUMP 1 **
DMP(14500,14650)      ** DUMP 2 **
CATLIST.
*
****************************************
* SEE OUTPUT NUMBER 3                  *
****************************************
*
PURGE(LOAD1,USELIB1,US,LIB2(USELIB3,DEMO)
CATLIST.
*
****************************************
* SEE OUTPUT NUMBER 4                  *
****************************************
*
R     7/8/9 END-OF-RECORD
  10  20  30  40  50
  60  70  80  90 100
R
ATTACH(DEMO=DEMOPF)
PURGE(LOAD1,USELIB1,USELIB2(USELIB3,DEMOPF)
Q     6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/07. 11.16.30.*11.16.25* PAGE    1

11.16.25.OVRLY2,T5.
11.16.25.UCCR, 72070,      0.024KCDS.
11.16.25.USER,DRF2755,.
11.16.25.MAP(PART)
11.16.25.GET,DEMO.
11.16.26.DEMO.
11.16.28.     END  MAIN
11.16.28.    022600 MAXIMUM EXECUTION FL.
11.16.28.     0.131 CP SECONDS EXECUTION TIME.
11.16.28.DMP(200)              ** DUMP 1 **
11.16.28.DMP(14500,14650)      ** DUMP 2 **
11.16.29.CATLIST.
11.16.29. CATLIST COMPLETE.
11.16.30.*
11.16.30.****************************************
11.16.30.* SEE OUTPUT NUMBER 3                  *
11.16.30.****************************************
11.16.30.*
11.16.30.PURGE(LOAD1,USELIB1,US,LIB2(USELIB3,DEMO)
11.16.30. LOAD1 NOT FOUND.
11.16.30.UEAD,       0.001KUNS.
11.16.30.UEPF,       0.060KUNS.
11.16.30.UEMS,       0.864KUNS.
11.16.30.UECF,       0.264SECS.
11.16.30.AESR,       1.437UNTS.
11.16.30.$OUT(*/OP=E)
11.16.30. NO FILES PROCESSED.
11.16.30.$DAYFILE(OUTPUT,JT=C)
11.16.51.UCLF, 72,073,       0.384KLNS.
```

```
 ENTERED MAIN
 CALL PRIM10
 ENTERED PRIM10
 ENTERED SEC11
 ENTERED SEC12
 CALL PRIM20
 ENTERED PRIM20
 ENTERED SEC21
 ITOTAL=              550
 THE END
10 20 30 40 50 60 70 80 90 100
```



```
CM DUMP FROM        0 TO      200.      VERSION = NOS 4K30-FNT101      81/10/07. 11.16.28.
     0      00000 00000 00000 00000      05160 42000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
     4      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
         DUPLICATED LINES.
    50      20120 00300 00000 07212      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
    54      56110 03110 00054 54710      51100 00001 03110 00055      64550 02550 00000 46000      00000 00000 00000 00000
    60      15051 52000 00000 00061      00000 15500 00000 00001      00000 00000 00000 00000      00000 00000 00000 00000
    64      04051 51700 00000 00000      40000 00000 00007 62754      40000 00000 01000 15001      40000 00000 40000 00000
    70      04051 51757 55000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
    74      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
   100      54000 00000 01000 00001      01462 20000 00000 14735      00000 00000 00000 00000      00000 00000 00000 00000
   104      00000 00000 00000 15033      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
   110      15011 11600 00000 04264      00000 00000 00000 00012      00000 00000 00000 00024      00000 00000 00000 00036
   114      00000 00000 00000 00050      00000 00000 00000 00062      00000 00000 00000 00074      00000 00000 00000 00106
   120      00000 00000 00000 00120      00000 00000 00000 00132      00000 00000 00000 00144      17252 42025 24000 00131
   124      04110 46000 00360 00174      00000 00000 00000 00174      00000 00000 00000 00174      03020 00000 01000 00176
   130      00000 00000 00000 00000      00000 00133 00000 00000      00000 00000 00000 00000      00000 00000 00000 00001
   134      00000 00000 00000 00000      00000 00044 00600 00000      00000 03762 03200 00000      00000 22600 00000 00000
   140      14000 00000 00000 06503      00000 00000 00000 00000      00000 00006 00000 04576      00000 00000 00000 01625
   144      23000 00000 00000 02003      00000 26702 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
   150      00000 00000 00000 00000      00004 00100 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
   154      00000 00000 00000 00000      00000 00000 00000 00000      00000 10000 00000 00000      00000 00000 00000 00000
   160      00000 00000 00000 00000      01040 00174 00000 00000      00000 00000 00000 00037      00000 00000 00000 00000
   164      00000 17400 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
   170      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00210
   174      34000 00000 00000 00000      55550 51624 05220 50455      15011 11600 00000 00000      55550 30114 14552 02211
   200      15343 30000 00000 00000      55550 51624 05220 50455      20221 11534 33000 00000      55550 51624 05220 50455
```

```
CM DUMP FROM   14500 TO   14650.      VERSION = NOS 4K30-FNT101      81/10/07. 11.16.29.
 14500      12661 01000 12003 46000      53120 21136 04000 14474      00000 17400 00000 00001      04000 14107 00000 00000
 14504      50207 77756 37621 54620      50307 77757 37632 13332      21673 15336 13632 54630      04000 14503 61000 46000
 14510      04000 14045 00000 00000      43666 03340 14520 37146      11161 50207 77761 73721      54720 50207 77762 74700
 14514      36771 37627 13227 21673      15226 13627 54620 63610      20736 01000 14474 46000      04000 14522 61000 46000
 14520      16146 63610 03010 14510      50207 77760 73621 54620      61700 00006 76160 46000      55207 36621 54620 67771
 14524      05700 14523 53320 21322      53430 36741 43302 46000      03740 14530 54740 43600      51600 14551 04000 14533
 14530      74740 20722 12767 46000      51700 14551 74730 20722      12773 54730 61000 46000      63720 75107 73220 73326
 14534      76710 37623 21673 13776      67701 73417 11464 36224      36334 46000 61000 46000      53420 10644 36227 53630
 14540      36337 73117 03110 14537      54006 51200 00065 43302      14222 74600 12763 53720      55201 53220 51300 14551
 14544      03030 14546 20252 10733      53720 46000 61000 46000      06600 14510 56311 56411      74700 20736 01000 14474
 14550      04000 14510 61000 46000      00000 00000 00000 00000      04000 13553 00000 00000      73410 50207 77770 74300
 14554      37623 51100 00065 14111      63411 63510 55101 37731      73771 63620 04000 14562      73771 21344 53130 46000
 14560      73210 21144 72310 00004      36773 53120 03120 14560      56340 63430 04400 14570      64730 77147 03630 14565
 14564      36771 04000 14562 46000      03330 14557 37316 13116      21373 15113 13616 46000      04000 14562 61000 46000
 14570      77365 27303 27707 24303      24707 44173 10266 46000      03040 14552 50307 77750      03130 14552 51300 14704
 14574      31631 03260 14552 76110      01000 14655 61000 46000      14111 53110 53010 46000      51100 14653 10611 46000
 14600      50607 77736 55111 46000      04000 14106 00000 00000      10611 51600 14652 74600      54661 51100 00065 73610
 14604      03360 14576 63560 14766      43652 15776 11661 12767      54710 10622 54661 43735      01000 14474 61000 46000
 14610      63710 43700 01000 14474      63610 53010 51200 00104      43752 11772 73220 76150      37612 13112 21673 15116
 14614      13212 12772 54720 76170      37612 72667 77715 46000      03260 14623 71100 14705      43652 15116 71602 00000
 14620      01000 12032 61000 46000      71600 41121 20650 43101      01000 12003 61000 46000      72220 02000 76770 37627
 14624      13727 21673 15776 13772      76260 37227 43666 46000      03220 14630 37776 11767      53070 20736 01000 14474
 14630      70607 77717 55601 55661      55661 55661 55661 55661      63460 76770 50707 77770      43700 50707 77767 46000
 14634      50707 77761 50707 77760      50707 77755 50707 77754      50707 77751 50707 77750      50707 77753 50707 77752
 14640      50707 77766 50707 77765      50707 77764 43673 46000      50607 77763 74600 46000      50607 77762 71600 00062
 14644      50607 77757 50607 77756      43102 74700 12717 56750      76651 20622 12616 56640      76740 76650 20622 12767
 14650      56751 51200 14654 46000      51100 00065 04000 14576      00000 00000 00003 43634      00000 00000 00000 02177
```

2

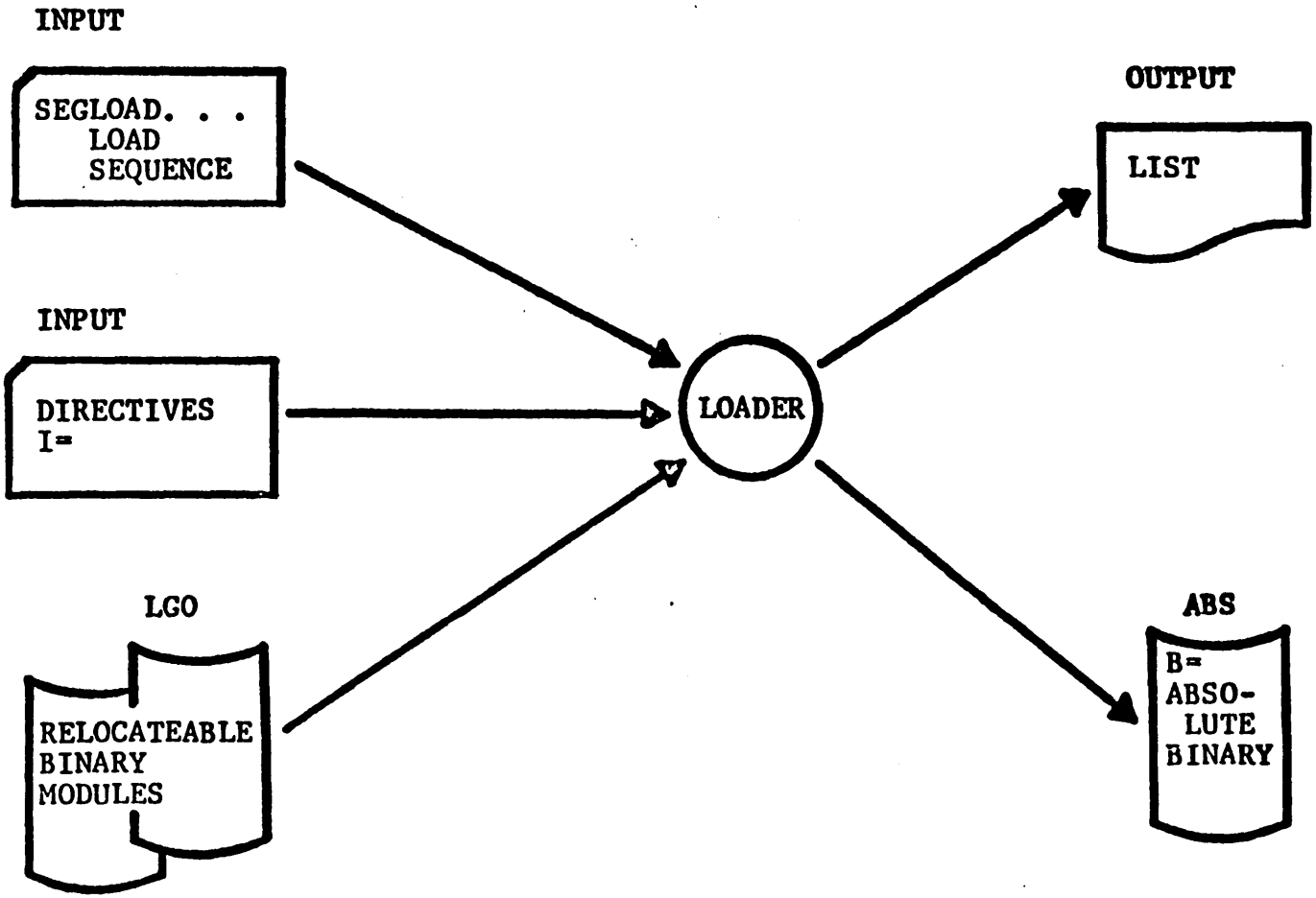


```
CATALOG CF  CRF2755          FM/SYS172  81/10/07. 11.16.29.  PAGE   1
```

3

```
INDIRECT ACCESS FILE(S)

 BUILD      M9315      M9354      M9452      M9551      M9652      M9751
 DEMO       M9316      M9355      M9471      M9552      M9671      M9752
 MTS9JOB    M9333      M9356      M9472      M9571      M9672      PAM
 M9000      M9334      M9357      M9501      M9591      M9701      RUN
 M9001      M9335      M9401      M9502      M9601      M9702      STLF
 M9286      M9336      M9402      M9521      M9602      M9721      TEST
 M9313      M9353      M9451      M9522      M9651      M9722      ZZZ
 M9314

DIRECT ACCESS FILE(S)

 QACQACR    STL

      50 INDIRECT ACCESS FILE(S),   TOTAL PRUS =      661.
       2 DIRECT ACCESS FILE(S),     TOTAL PRUS =      298.
```

SEGLOAD(I=lfn$_1$,B=lfn$_2$)



```
SEG1,T5.
USER,DRF2755,DRF.
FTN,R=0.
MAP(PART)
SEGLOAD.
LGO.
R     7/8/9 END-OF-RECORD
      PROGRAM MAIN(OUTPUT)
      COMMON /L/ LIST(11)
      DATA LIST /0,1,1,0,0,1,1,0,1,0,1/
      PRINT 100
  100 FORMAT(1H1)
      DO 20 I=1,10
      IF(LIST(I) .EQ. 0) CALL ZERO(I)
      IF(LIST(I) .EQ. 1) CALL ONE(I)
   20 CONTINUE
      END
      SUBROUTINE ZERO(I)
      COMMON /L/ LIST(11) /ZERO/ LOC
      DATA LOC /0/
      X=SQRT(0.)
      IF(LOC .NE. 0) GO TO 10
      LOC = 1
      PRINT 1,I,LIST(I)
    1 FORMAT(* STEP*I3*, LIST VALUE =*,I2,3X,*ZERO WAS LOADED*)
      RETURN
   10 PRINT 2,I,LIST(I)
    2 FORMAT(* STEP*I3*, LIST VALUE =*,I2,3X,*ZERO WAS REENTERED*)
      RETURN
      END
      SUBROUTINE ONE(I)
      COMMON /L/ LIST(11) /ONE/ LOC
      DATA LOC /0/
      X=SIN(1.0)
      IF(LOC .NE. 0) GO TO 10
      LOC = 1
      PRINT 1, I, LIST(I)
    1 FORMAT(* STEP*I3*, LIST VALUE =*,I2,3X,*ONE WAS LOADED*)
      RETURN
   10 PRINT 2, I, LIST(I)
    2 FORMAT(* STEP*I3*, LIST VALUE =*,I2,3X,*ONE WAS REENTERED*)
      RETURN
      END
R     7/8/9 END-OF-RECORD
   TREE   MAIN-(ZERO,ONE)
   GLOBAL L
   END
Q     6/7/8/9 END-OF-INFORMATION
```

```
    FULL  DAYFILE.  81/10/07. 11.17.06.*11.16.53* PAGE    1

11.16.53.SEG1,T5.
11.16.53.UCCR, 72070,      0.045KCDS.
11.16.54.USER,DRF2755,.
11.16.54.FTN,R=0.                                          1
11.16.57.      .341 CP SECONDS COMPILATION TIME
11.16.57.MAP(PART)
11.16.58.SEGLOAD.                                          2
11.16.58.LGO.
11.17.05.      END  MAIN
11.17.05.     015000 MAXIMUM EXECUTION FL.
11.17.05.      0.081 CP SECONDS EXECUTION TIME.
11.17.05.UEAD,       0.091KUNS.
11.17.05.UEMS,       4.462KUNS.
11.17.05.UECP,       1.865SECS.
11.17.05.AESR,       3.956UNTS.
11.17.05.$OUT(*/OP=F)
11.17.05. NO FILES PROCESSED.
11.17.06.$DAYFILE(OUTPUT,JT=D)
11.17.24.UCLP, 72,073,      0.576KLNS.
```

```
     PROGRAM MAIN       73/74   CPT=1                    FTN 4.8+552        81/10/07. 11.16.54       PAGE      1

                                                                  1
 1                PROGRAM MAIN(OUTPUT)
                  COMMON /L/ LIST(11)
                  DATA LIST /0,1,1,0,0,1,1,0,1,0,1/
                  PRINT 100
 5              100 FORMAT(1H1)
                  DO 20 I=1,10
                  IF(LIST(I) .EQ. 0) CALL ZERO(I)
                  IF(LIST(I) .EQ. 1) CALL ONE(I)
                20 CONTINUE
10                END
```

```
  SUBROUTINE ZERO       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.54       PAGE      1

 1                SUBROUTINE ZERO(I)
                  COMMON /L/ LIST(11) /ZERO/ LOC
                  DATA LOC /0/
                  X=SQRT(0.)
 5                IF(LOC .NE. 0) GO TO 10
                  LOC = 1
                  PRINT 1,I,LIST(I)
                1 FORMAT(* STEP*I3*, LIST VALUE =*,I2,3X,*ZERO HAS LOADED*)
                  RETURN
10               10 PRINT 2,I,LIST(I)
                 2 FORMAT(* STEP*I3*, LIST VALUE =*,I2,3X,*ZERO HAS REENTERED*)
                  RETURN
                  END
```

```
  SUBROUTINE ONE        73/74   CPT=1                    FTN 4.8+552        81/10/07. 11.16.54       PAGE      1

 1                SUBROUTINE ONE(I)
                  COMMON /L/ LIST(11) /ONE/ LOC
                  DATA LOC /0/
                  X=SIN(1.0)
 5                IF(LOC .NE. 0) GO TO 10
                  LOC = 1
                  PRINT 1, I, LIST(I)
                1 FORMAT(* STEP*I3*, LIST VALUE =*,I2,3X,*ONE HAS LOADED*)
                  RETURN
10               10 PRINT 2, I, LIST(I)
                 2 FORMAT(* STEP*I3*, LIST VALUE =*,I2,3X,*ONE HAS REENTERED*)
                  RETURN
                  END
```

SEGLOAD DIRECTIVES.

2

```
   TREE   MAIN-(ZERO,ONE)
   GLOBAL L
   END
```

TREE DIAGRAM.

```
*MAIN
 +
 +*ZERO
 +
 +*ONE
```

FWA OF THE LOAD 1006
LWA+1 OF THE LOAD 10671

WRITTEN TO FILE ABS

TRANSFER ADDRESS -- MAIN 3346

8-05

------ SEGMENT - MAIN

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| (MAIN) | | 1006 | 2 | | | | | | |
| /FCL.C./ | GS | 1010 | 30 | | | | | | |
| /08.IO./ | GS | 1040 | 130 | | | | | | |
| /STP.END/ | GS | 1170 | 1 | | | | | | |
| /FCL=ENT/ | GS | 1171 | 43 | | | | | | |
| /L/ | G | 1234 | 13 | | | | | | |
| /FDL.COM/ | | 1247 | 14 | | | | | | |
| MAIN | | 1263 | 2112 | LGO | 81/10/07 | FTN | 4.8 552 | 666X I | PROGRAM OPT=1 |
| EXP.MSG | | 3375 | 16 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 552 | | COMMON ERROR MESSAGES FOR EXPONENTIATION. |
| SYSAID= | | 3413 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| SYS=1ST | | 3414 | 70 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 552 | | MATH LIBRARY LINK TO ERROR MESSAGE PROCESSOR. |
| O2NTRY= | | 3504 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 552 | | FCL INITIALIZATION ROUTINE. |
| COMIC= | | 3504 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FCL | | 3537 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FCL CAPSULE LOADING |
| FECHSK= | | 3577 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | | 3640 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | CONVERTED DATA STORAGE |
| FLTCUT= | | 3643 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON FLOATING OUTPUT CODE |
| FMTAP= | | 4154 | 357 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | | 4533 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | | 5234 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | FCL MISC. UTILITIES. |
| GETFIT= | | 5261 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | | 5355 | 470 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | OUTPUT FORMAT INTERPRETER. |
| OUTCOM= | | 6045 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 554 | | COMMON OUTPUT CODE |

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|---|
| OUTC= | | 6221 | 162 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORMATTED WRITE FORTRAN RECORD. |
| CPU.CPM | | 6403 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| CPU.SYS | | 6410 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| FDL.RES | | 6450 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | | 6661 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | | 7103 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | | 7540 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| ERR$RM | | 7617 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | | 7644 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | | 7733 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | | 7740 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | | 10122 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | | 10130 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | | 10144 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | | 10202 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | | 10224 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | | 10233 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | | 10441 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |
| CMF.SDA | | 10463 | 47 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SET DABA. |

8+551

8+------ SEGMENT - ZERO

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|---|
| /L/ | S | 1234 | 13 | | | | | | | |
| (ZERO) | | 10532 | 0 | | | | | | | |
| /ZERO/ | | 10532 | 1 | | | | | | | |
| ZERO | | 10533 | 56 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SQRT. | | 10611 | 32 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | COMPUTE THE SQUARE ROOT OF X. OPT=ALL. |

-------- SEGMENT - ONE

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|---|
| /L/ | S | 1234 | 13 | | | | | | | |
| (ONE) | | 10532 | 0 | | | | | | | |
| /ONE/ | | 10532 | 1 | | | | | | | |
| ONE | | 10533 | 56 | LGO | 81/10/07 | FTN | 4.8 | 552 | 666X I | SUBROUTINEOPT=1 |
| SINCOS. | | 10611 | 60 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | TRIGONOMETRIC SINE OR COSINE OF X. OPT=ALL. |

1.383 CP SECONDS  264008 CM STORAGE USED  5 TABLE MOVES

NO. OF SEGMENTS+PROGRAMS+BLOCKS USED = 77  MAXIMUM PERMITTED = 8192

```
STEP  1, LIST VALUE = 0   ZERO WAS LOADED
STEP  2, LIST VALUE = 1   ONE WAS LOADED
STEP  3, LIST VALUE = 1   ONE WAS REENTERED
STEP  4, LIST VALUE = 0   ZERO WAS LOADED
STEP  5, LIST VALUE = 0   ZERO WAS REENTERED
STEP  6, LIST VALUE = 1   ONE WAS LOADED
STEP  7, LIST VALUE = 1   ONE WAS REENTERED
STEP  8, LIST VALUE = 0   ZERO WAS LOADED
STEP  9, LIST VALUE = 1   ONE WAS LOADED
STEP 10, LIST VALUE = 0   ZERO WAS LOADED
```

```
SEG2,T5.
USER,DRF2755,DRF.
FTN,B=BIN,R=0.
MAP(PART)
SEGLOAD,B=SEGCHK.
BIN.
SAVE(SEGCHK=SEG)
EXIT.
DEFINE(SEGBIN)
REWIND(SEGCHK)
COPY(SEGCHK,SEGBIN,VERIFY)
R     7/8/9 END-OF-RECORD
      PROGRAM ROOT(OUTPUT,TAPE6=OUTPUT)
      WRITE(6,100)
  100 FORMAT(1H1)
      WRITE (6,*) "ROOT IS IN AND CALLING CHECK"
      CALL CHECK
      WRITE (6,*) "ROOT IS BACK AND CALLING J1"
      CALL J1(4HROOT)
      WRITE (6,*) "ROOT IS BACK AND CALLING C1"
      CALL C1
      WRITE (6,*) "ROOT IS BACK AND CALLING B11"
      CALL B11
      WRITE (6,*) "ROOT IS BACK AND CALLING C1"
      CALL C1
      WRITE (6,*) "ROOT IS BACK AND TERMINATING"
      END
      SUBROUTINE B11
      WRITE (6,*) " B11 IS IN AND CALLING J1"
      CALL J1(3HB11)
      WRITE (6,*) " B11 IS BACK AND CALLING G1"
      CALL G1
      WRITE (6,*) " B11 IS BACK AND CALLING H1"
      CALL H1
      WRITE (6,*) " B11 IS BACK AND RETURNING TO ROOT"
      END
      SUBROUTINE G1
      WRITE (6,*) " G1 IS IN AND RETURNING TO B11"
      END
      SUBROUTINE H1
      WRITE (6,*) " H1 IS IN AND RETURNING TO B11"
      END
      SUBROUTINE C1
      COMMON /LABC/ LABC
      DATA LABC /5HFIRST/
      WRITE (6,*) " C1 IS IN AND CALLING J1"
      CALL J1(2HC1)
      WRITE (6,*) " C1 IS BACK AND CALLING D1"
      CALL D1
      WRITE (6,*) " C1 IS BACK AND RETURNING TO ROOT"
      END
      SUBROUTINE D1
      WRITE (6,*) " D1 IS IN AND CALLING E1"
      CALL E1
      WRITE (6,*) " D1 IS BACK AND CALLING F1"
      CALL F1
      WRITE (6,*) " D1 IS BACK AND RETURNING TO C1"
      END
      SUBROUTINE E1
      COMMON /LCBA/ LABC
      WRITE (6,*) " E1 IS IN AND RETURNING TO D1 -", LABC,"CALL"
      LABC=6HSECOND
      END
      SUBROUTINE F1
      WRITE (6,*) " F1 IS IN AND RETURNING TO D1"
      END
      SUBROUTINE CHECK
      WRITE (6,*) " CHECK IS IN AND RETURNING"
      END
      SUBROUTINE J1(ITEM)
      WRITE (6,*) " J1 IS IN AND RETURNING TO", ITEM
      END
R     7/8/9 END-OF-RECORD
 FIRST     TREE   B11-(G1,H1)
 SECOND    TREE   C1-(F1,F1)
           TREE   ROOT-(FIRST,SECOND)
*                 EQUATE LABEL COMMON BLOCKS LABC AND LCBA
 LABC      EQUAL  LCBA
*                 FORCE CHECK INTO SEGMENT C1,
*                 OTHERWISE CHECK WOULD BE LOADED IN ROOT.
 C1        INCLUDE CHECK,C1
*                 MAKE LABC/ GLOBAL AND FORCE IT
*                 INTO SEGMENT C1.  -SAVE- WILL CAUSE
*                 /LABC/ TO BE SAVED WHENEVER C1 IS
*                 OVERLAYED.
 C1        GLOBAL LABC-SAVE
*                 MAKE J1 DIRECTLY ACCESSIBLE FROM
*                 ALL ROUTINES AT THE LOWER LEVEL.
           LEVEL
           TREE   J1
           END
0     6/7/8/9 END-OF-INFORMATION
```

```
      FULL  DAYFILE.  81/10/07. 11.17.20.*11.16.58* PAGE    1

11.16.58.SEG2,T5.
11.16.58.UCCR, 72070,       0.089KCDS.
11.16.59.USER,DRF2755,.
11.16.59.FTN,B=BIN,R=0.
11.17.10.       .563 CP SECONDS COMPILATION TIME
11.17.10.MAP(PART)
11.17.11.SEGLOAD,B=SEGCHK.
11.17.11.BIN.
11.17.18.      END  ROOT
11.17.18.     015600 MAXIMUM EXECUTION FL.
11.17.18.       0.102 CP SECONDS EXECUTION TIME.
11.17.19.SAVE(SEGCHK=SEG)
11.17.19.EXIT.
11.17.19.UEAC,       0.001KUNS.
11.17.19.UEPF,       0.033KUNS.
11.17.19.UEMS,       8.301KUNS.
11.17.19.UECP,       2.328SECS.
11.17.19.AESR,       5.349UNTS.
11.17.20.$OUT(*/OP=F)
11.17.20. NO FILES PROCESSED.
11.17.20.$DAYFILE(OUTPUT,JT=D)
11.17.53.UCLP, 72,073,      1.088KLNS.
```

1

2

```
     PROGRAM ROOT        73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE     1

1                 PROGRAM ROOT(OUTPUT,TAPE6=OUTPUT)
                  WRITE(6,100)                                    1
              100 FORMAT(1H1)
                  WRITE (6,*) #ROOT IS IN AND CALLING CHECK#
5                 CALL CHECK
                  WRITE (6,*) #ROOT IS BACK AND CALLING J1#
                  CALL J1(4HROOT)
                  WRITE (6,*) #ROOT IS BACK AND CALLING C1#
                  CALL C1
10                WRITE (6,*) #ROOT IS BACK AND CALLING B11#
                  CALL B11
                  WRITE (6,*) #ROOT IS BACK AND CALLING C1#
                  CALL C1
                  WRITE (6,*) #ROOT IS BACK AND TERMINATING#
15                END
```

```
     SUBROUTINE B11      73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE     1

1                 SUBROUTINE B11
                  WRITE (6,*) # B11 IS IN AND CALLING J1#
                  CALL J1(3HB11)
                  WRITE (6,*) # B11 IS BACK AND CALLING G1#
5                 CALL G1
                  WRITE (6,*) # B11 IS BACK AND CALLING H1#
                  CALL H1
                  WRITE (6,*) # B11 IS BACK AND RETURNING TO ROOT#
                  END
```

```
     SUBROUTINE G1       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE     1

1                 SUBROUTINE G1
                  WRITE (6,*) # G1 IS IN AND RETURNING TO B11#
                  END
```

```
     SUBROUTINE H1       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE     1

1                 SUBROUTINE H1
                  WRITE (6,*) # H1 IS IN AND RETURNING TO B11#
                  END
```

```
SUBROUTINE C1          73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE      1

1               SUBROUTINE C1
                COMMON /LABC/ LABC
                DATA LABC /5HFIRST/
                WRITE (6,*) ' C1 IS IN AND CALLING J1'
5               CALL J1(2HC1)
                WRITE (6,*) ' C1 IS BACK AND CALLING D1'
                CALL D1
                WRITE (6,*) ' C1 IS BACK AND RETURNING TO ROOT'
                END


SUBROUTINE D1          73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE      1

1               SUBROUTINE D1
                WRITE (6,*) ' D1 IS IN AND CALLING E1'
                CALL E1
                WRITE (6,*) ' C1 IS BACK AND CALLING F1'
5               CALL F1
                WRITE (6,*) ' D1 IS BACK AND RETURNING TO C1'
                END


SUBROUTINE E1          73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE      1

1               SUBROUTINE E1
                COMMON /LCBA/ LABC
                WRITE (6,*) ' E1 IS IN AND RETURNING TO D1 -', LABC,'CALL'
                LABC=6HSECCND
5               END


SUBROUTINE F1          73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE      1

1               SUBROUTINE F1
                WRITE (6,*) ' F1 IS IN AND RETURNING TO D1'
                END


SUBROUTINE CHECK       73/74   OPT=1                    FTN 4.8+552        81/10/07. 11.16.59        PAGE      1

1               SUBROUTINE CHECK
                WRITE (6,*) ' CHECK IS IN AND RETURNING'
                END


SUBROUTINE J1          73/74   OPT=1

1               SUBROUTINE J1(ITEM)
                WRITE (6,*) ' J1 IS IN AND RETURNING TO', I
                END
```

```
LOAD MAP - SEGMENTED LOAD.                    CYBER LOADER 1.5-552      81/10/07. 11.17.11.      PAGE      1


SEGLOAD DIRECTIVES.

 FIRST    TREE   B11-(G1,H1)
 SECOND   TREE   C1-(E1,F1)
          TREE   ROOT-(FIRST,SECOND)
*                EQUATE LABEL COMMON BLOCKS LABC AND LCBA
 LABC     EQUAL  LCBA
*                FORCE CHECK INTO SEGMENT C1,
*                OTHERWISE CHECK WOULD BE LOADED IN ROOT.
 C1       INCLUDE CHECK,C1
*                MAKE LABC/ GLOBAL AND FORCE IT
*                INTO SEGMENT C1.  -SAVE- WILL CAUSE
*                /LABC/ TO BE SAVED WHENEVER C1 IS
*                OVERLAYED.
 C1       GLOBAL LABC-SAVE
*                MAKE J1 DIRECTLY ACCESSIBLE FROM
*                ALL ROUTINES AT THE LOWER LEVEL.
          LEVEL
          TREE   J1
          END


TREE DIAGRAM.

*ROOT
 +
 +*B11
 + +
 + +*G1
 + +
 + +*H1
 +
 +*C1
   +
   +*E1
   +
   +*F1

================================================================================

*J1


FWA OF THE LOAD           1020
LWA+1 OF THE LOAD        11353

WRITTEN TO FILE     SEGCHK

TRANSFER ADDRESS -- ROOT           3351


-------- SEGMENT - ROOT
```

2

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER | LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|---|
| (ROOT) | | 1020 | 5 | | | | | | | |
| /FCL.C./ | GS | 1025 | 30 | | | | | | | |
| /Q8.IC./ | GS | 1055 | 130 | | | | | | | |
| /STP.ENT/ | GS | 1205 | 1 | | | | | | | |
| /FCL=ENT/ | GS | 1206 | 43 | | | | | | | |
| /FCL.COM/ | | 1251 | 14 | | | | | | | |
| ROOT | | 1265 | 2175 | BIN. | 81/10/07 | FTN | 4.8 | 552 | 666X I | PROGRAM OPT=1 |
| SYSAID= | | 3462 | 1 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | LINK BETWEEN SYS=AID AND INITIALIZATION CODE. |
| Q2NTRY= | | 3463 | 0 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 552 | | FCL INITIALIZATION ROUTINE. |
| COMIO= | | 3463 | 33 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON CODED I/O ROUTINES AND CONSTANTS. |
| FCL=FCL | | 3516 | 40 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL CAPSULE LOADING |
| FECMSK= | | 3556 | 41 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | INITIALIZE CONSTANTS. |
| FEIFST= | | 3617 | 3 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CONVERTED DATA STORAGE |
| FLTOUT= | | 3622 | 311 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON FLOATING OUTPUT CODE |
| FMTAP= | | 4133 | 357 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | CRACK APLIST AND FORMAT FOR KODER/KRAKER. |
| FORSYS= | | 4512 | 501 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORTRAN OBJECT LIBRARY UTILITIES. |
| FORUTL= | | 5213 | 25 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FCL MISC. UTILITIES. |
| GETFIT= | | 5240 | 74 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LOCATE A FIT GIVEN A FILE DESCRIPTOR |
| KODER= | | 5334 | 470 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | OUTPUT FORMAT INTERPRETER. |
| LOOUT= | | 6024 | 244 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT FORMATTING |
| OUTCOM= | | 6270 | 154 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | COMMON OUTPUT CODE |
| OUTO= | | 6444 | 162 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | FORMATTED WRITE FORTRAN RECORD. |
| OUTF= | | 6626 | 204 | SL-FORTRAN | 81/08/10 | COMPASS | 3.6 | 554 | | LIST DIRECTED OUTPUT CONTROL |
| CPU.CPP | | 7032 | 5 | SL-SYSLIB | 81/10/05 | COMPASS | 3.6 | 552 | | 79/05/10. 81/09/17. CONTROL POINT MANAGER PROCE |
| CPU.SYS | | 7037 | 40 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | PROCESS SYSTEM REQUEST. |
| FDL.RES | | 7077 | 211 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FAST DYNAMIC LOADER RESIDENT. |
| FDL.MMI | | 7310 | 222 | SL-SYSLIB | 80/09/24 | COMPASS | 3.6 | 528 | | FDL MEMORY MANAGER INTERFACE. |
| CTL$RM | | 7532 | 435 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLING ROUTINE. |
| CTL$SKP | | 10167 | 57 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM CONTROLLER - SKIP PHYSICAL/FILE. |
| ERR$RM | | 10246 | 25 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM ERROR PROCESSOR ENTRY. |
| LIST$RM | | 10273 | 67 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - ALLOCATE SPACE FOR LIST OF FILES |
| RM$SYS= | | 10362 | 5 | SL-SYSLIB | 80/09/25 | COMPASS | 3.6 | 528 | | CRM - POST RA+1 REQUEST |
| CMF.ALF | | 10367 | 162 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - ALLOCATE FIXED. |
| CMF.CSF | | 10551 | 6 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - CHANGE SPECS FIXED. |
| CMM.FFA | | 10557 | 14 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FIXED FREE ALGORITHM. |
| CMF.FRF | | 10573 | 36 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - FREE FIXED. |
| CMF.GSS | | 10631 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - GET SUMMARY STATISTICS. |
| CMM.MEM | | 10653 | 7 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | |
| CMM.R | | 10662 | 206 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - RESIDENT SUBROUTINES. |
| CMF.SLF | | 11070 | 22 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SHRINK AT LWA FIXED. |
| CMF.SCA | | 11112 | 47 | SL-SYSLIB | 80/11/25 | COMPASS | 3.6 | 528 | | CMM V1.1 - SET DABA. |



-------- SEGMENT - B11

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| (P11) | | 11161 | 3 | | | | | |
| B11 | | 11164 | 60 | BIN | 81/10/07 | FTN 4.8 552 | 666X I | SUBROUTINEOPT=1 |

-------- SEGMENT - G1

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| (G1) | | 11244 | 0 | | | | | |
| G1 | | 11244 | 14 | BIN | 81/10/07 | FTN 4.8 552 | 666X I | SUBROUTINEOPT=1 |

-------- SEGMENT - H1



PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| (H1) | | 11244 | 0 | | | | | |
| H1 | | 11244 | 14 | BIN | 81/10/07 | FTN 4.8 552 | 666X I | SUBROUTINEOPT=1 |

-------- SEGMENT - C1

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| (C1) | | 11161 | 3 | | | | | |
| /LABC/ | CS | 11164 | 1 | | | | | |
| CHECK | | 11165 | 14 | BIN | 81/10/07 | FTN 4.8 552 | 666X I | SUBROUTINEOPT=1 |
| C1 | | 11201 | 46 | BIN | 81/10/07 | FTN 4.8 552 | 666X I | SUBROUTINEOPT=1 |
| D1 | | 11247 | 41 | BIN | 81/10/07 | FTN 4.8 552 | 666X I | SUBROUTINEOPT=1 |

-------- SEGMENT - F1

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|---|
| /LOBA/ | S | 11164 | 1 | | | | | | |
| (E1) | | 11310 | 0 | | | | | | |
| E1 | | 11310 | 23 | BIN | 81/10/07 | FTN | 4.8 552 | 666X I | SUBROUTINEOPT=1 |

-------- SEGMENT - F1

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| (F1) | 11310 | 0 | | | | | | |
| F1 | 11310 | 14 | BIN | 81/10/07 | FTN | 4.8 552 | 666X I | SUBROUTINEOPT=1 |

-------- SEGMENT - J1

PROGRAM AND BLOCK ASSIGNMENTS.

| BLOCK | ADDRESS | LENGTH | FILE | DATE | PROCSSR | VER LEVEL | HARDWARE | COMMENTS |
|---|---|---|---|---|---|---|---|---|
| (J1) | 11333 | 0 | | | | | | |
| J1 | 11333 | 20 | BIN | 81/10/07 | FTN | 4.8 552 | 666X I | SUBROUTINEOPT=1 |

1.598 CP SECONDS  272008 CM STORAGE USED  8 TABLE MOVES

NO. OF SEGMENTS+PROGRAMS+BLOCKS USED = 88  MAXIMUM PERMITTED = 8192

```
≠ROOT IS IN ANC CALLING CHECK≠
≠ CHECK IS IN AND RETURNING≠
≠ROOT IS BACK AND CALLING J1≠
≠ J1 IS IN AND RETURNING TO≠ ≠RCOT       ≠
≠ROOT IS BACK AND CALLING C1≠
≠ C1 IS IN AND CALLING J1≠
≠ J1 IS IN ANC RETURNING TO≠ ≠C1         ≠
≠ C1 IS BACK AND CALLING D1≠
≠ D1 IS IN AND CALLING E1≠
≠ E1 IS IN AND RETURNING TO D1 -≠ ≠FIRST      ≠ ≠CALL≠
≠ D1 IS BACK ANC CALLING F1≠
≠ F1 IS IN AND RETURNING TO D1≠
≠ C1 IS BACK ANC RETURNING TO C1≠
≠ C1 IS BACK AND RETURNING TO RCOT≠
≠ROOT IS BACK AND CALLING B11≠
≠ B11 IS IN ANC CALLING J1≠
≠ J1 IS IN AND RETURNING TO≠ ≠B11        ≠
≠ B11 IS BACK AND CALLING G1≠
≠ G1 IS IN AND RETURNING TO B11≠
≠ B11 IS BACK AND CALLING H1≠
≠ H1 IS IN AND RETURNING TO B11≠
≠ B11 IS BACK AND RETURNING TO ROOT≠
≠ROOT IS BACK AND CALLING C1≠
≠ C1 IS IN AND CALLING J1≠
≠ J1 IS IN AND RETURNING TO≠ ≠C1         ≠
≠ C1 IS BACK AND CALLING D1≠
≠ D1 IS IN AND CALLING E1≠
≠ E1 IS IN AND RETURNING TO D1 -≠ ≠SECCND     ≠ ≠CALL≠
≠ D1 IS BACK AND CALLING F1≠
≠ F1 IS IN AND RETURNING TO D1≠
≠ D1 IS BACK ANC RETURNING TO C1≠
≠ C1 IS BACK AND RETURNING TO RCOT≠
≠ROOT IS BACK AND TERMINATING≠
```

```
SEG3,T5.
USER,DRF2755,DRF.
GET,SEG.
MAP(ON)
SEG.
DMP(1100)          ** DUMP NUMBER 1 **
PURGE,SEG.
CATLIST.
*
****************************************
* SEE OUTPUT NUMBER 2                  *
****************************************
*
R
ATTACH(SEG=SEGBIN)
PURGE(SEGBIN)
Q     6/7/8/9 END-OF-INFORMATION
```

```
          FULL  DAYFILE.  81/10/07. 11.17.40.*11.17.35* PAGE    1

11.17.35.SEG3,T5.
11.17.35.UCCR, 72070,      0.015KCDS.
11.17.35.USER,DRF2755,.
11.17.35.GET,SEG.
11.17.36.MAP(ON)
11.17.36.SEG.
11.17.38.      END  ROOT
11.17.38.     015600 MAXIMUM EXECUTION FL.
11.17.38.      0.101 CP SECONDS EXECUTION TIME.
11.17.38.DMP(1100)          ** DUMP NUMBER 1 **
11.17.39.PURGE,SEG.
11.17.39.CATLIST.
11.17.39. CATLIST COMPLETE.
11.17.39.*
11.17.39.****************************************
11.17.39.* SEE OUTPUT NUMBER 2                  *
11.17.39.****************************************
11.17.39.*
11.17.39.UEAD,       0.001KUNS.
11.17.39.UEPF,       0.048KUNS.
11.17.39.UEMS,       0.841KUNS.
11.17.39.UECP,       0.256SECS.
11.17.39.AESR,       1.394UNTS.
11.17.39.$OUT(*/OP=E)
11.17.40. NO FILES PROCESSED.
11.17.40.$DAYFILE(OUTPUT,JT=D)
11.18.12.UCLP, 72,073,      0.448KLNS.
```

```
≠ROOT IS IN AND CALLING CHECK≠
≠ CHECK IS IN AND RETURNING≠
≠ROOT IS BACK AND CALLING J1≠
≠ J1 IS IN AND RETURNING TO≠ ≠RCOT       ≠
≠ROOT IS BACK AND CALLING C1≠
≠ C1 IS IN AND CALLING J1≠
≠ J1 IS IN AND RETURNING TO≠ ≠C1         ≠
≠ C1 IS BACK AND CALLING D1≠
≠ D1 IS IN AND CALLING E1≠
≠ E1 IS IN AND RETURNING TO D1 -≠ ≠FIRST      ≠ ≠CALL≠
≠ D1 IS BACK AND CALLING F1≠
≠ F1 IS IN AND RETURNING TO D1≠
≠ D1 IS BACK AND RETURNING TO C1≠
≠ C1 IS BACK AND RETURNING TO RCOT≠
≠ROOT IS BACK AND CALLING B11≠
≠ B11 IS IN AND CALLING J1≠
≠ J1 IS IN AND RETURNING TO≠ ≠B11        ≠
≠ B11 IS BACK AND CALLING G1≠
≠ G1 IS IN AND RETURNING TO B11≠
≠ B11 IS BACK AND CALLING H1≠
≠ H1 IS IN AND RETURNING TO B11≠
≠ B11 IS BACK AND RETURNING TO ROOT≠
≠ROOT IS BACK AND CALLING C1≠
≠ C1 IS IN AND CALLING J1≠
≠ J1 IS IN AND RETURNING TO≠ ≠C1         ≠
≠ C1 IS BACK AND CALLING D1≠
≠ D1 IS IN AND CALLING E1≠
≠ E1 IS IN AND RETURNING TO D1 -≠ ≠SECOND     ≠ ≠CALL≠
≠ D1 IS BACK AND CALLING F1≠
≠ F1 IS IN AND RETURNING TO D1≠
≠ D1 IS BACK AND RETURNING TO C1≠
≠ C1 IS BACK AND RETURNING TO RCOT≠
≠ROOT IS BACK AND TERMINATING≠
```

```
CM DUMP FROM       0 TO    1100.      VERSION = NOS 4K30-FNT101      81/10/07. 11.17.38.
     0      00000 00000 00000 00000      05160 42000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
     4      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
         DUPLICATED LINES.
    64      23050 70000 00000 00000      40000 00000 00007 66424      40000 00000 01000 00100      40000 00000 40000 00000
    70      23050 75700 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
    74      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
   100      54000 00000 00000 00001      00000 00000 00000 11161      00000 00000 00000 00000      00000 00000 00000 00000
   104      00000 00000 00000 11353      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
   110      22171 72400 00000 03351      23050 70000 00000 00023      04124 00000 00030 11310      00000 00000 00000 11324
   114      00000 00000 00000 11310      02770 00000 01000 11325      00000 00000 00000 00000      00000 00113 00000 00000
   120      00000 00000 10000 01241      32323 23232 36350 00023      04114 00000 00030 11164      00000 00000 00000 11165
   124      00000 00000 00000 11164      03050 00000 00000 11166      00000 00000 00000 00000      00000 00002 00000 00000
   130      00000 00000 00100 00111      00000 00000 00000 00020      00000 00000 00000 11310      00000 00000 00000 11353
   134      00000 00000 00000 11333      00000 11200 00000 00001      00011 25600 00000 00000      00000 00000 10750 00001
   140      00000 00000 10540 02767      00000 00000 11220 01526      00000 00001 24230 00011      00000 00000 10560 00012
   144      00000 00000 10540 07024      00000 00000 11070 00164      11040 01526 00002 26000      00000 00000 67660 00000
   150      04000 11256 00000 00000      00000 00000 00000 01054      00000 00000 00000 21556      00000 00000 00000 00000
   154      55000 00000 00000 00000      20110 00000 00000 01054      55161 71646 05300 50325      24010 21405 55271 72204
   160      55141 70104 11160 75501      55230 50715 05162 45700      00000 00080 00000 00000      04000 11257 00000 00000
   164      01000 00331 61000 46000      61700 00670 01000 00247      21622 20601 53167 73110      03310 00172 01000 00443
   170      01000 00276 61000 46000      66400 01000 00175 46000      71603 01222 27606 20652      71100 00136 12661 46000
   174      01000 00633 61000 46000      04000 00172 00000 00000      04430 00175 56174 73610      61440 00002 03260 00176
   200      43653 15616 63560 21160      43760 15717 03070 00176      63675 66200 61000 46000      43014 56150 11101 46000
   204      03010 00207 66221 46000      66551 05560 00203 46000      04000 00176 61000 46000      76620 51600 00245 66200
   210      56152 66221 73610 46000      03260 00210 21122 20101      53217 73220 03220 00215      64510 51100 00245 63212
   214      04000 00205 61000 46000      43001 20022 51500 00244      05210 00223 54110 15610      53260 10622 54610 43014
   220      11606 03060 00240 43715      20736 11771 74610 12776      12775 54720 04000 00213      51100 00245 20122 56250
   224      73620 54611 53220 10622      56650 56351 15330 73330      12613 12665 54620 43714      11272 03020 00240 20736
   230      66551 56150 15210 53220      11371 10622 56650 73110      20236 11272 03020 00240      03310 00236 56451 15440
   234      73440 12643 12665 54620      04000 00230 61000 46000      51400 00246 12643 12665      54620 04000 00213 46000
   240      71100 00156 43652 15116      76600 01000 00662 46000      71600 41121 20650 43101      01000 00633 61000 46000
   244      01000 00163 00000 00000      00000 00000 00000 00002      00000 00000 00000 03356      04000 00166 00000 00000
   250      43001 74610 20022 73210      03220 00255 15310 53230      10722 54710 43715 20736      11171 12616 54620 46000
   254      04000 00247 61000 46000      37726 03370 00257 73620      53160 73210 04000 00255      66600 51100 00131 63410
   260      56167 73710 03270 00263      15110 73710 37572 46000      03250 00264 63370 46000      61660 00002 05640 00260
   264      53120 43414 20436 11641      21622 53363 10733 53720      73720 73610 53160 11541      12775 54730 61000 46000
   270      37762 03370 00274 53160      54331 73260 73610 10733      54710 11741 12772 54730      04000 00270 61000 46000
   274      54170 12610 54610 46000      04000 00247 61000 46000      04000 00171 00000 00000      56127 66420 54211 73110
   300      73220 36612 51100 00133      51200 00134 51300 00132      37221 03320 00304 73730      10133 54710 61000 46000
   304      37216 21274 15112 11626      12616 01000 00540 46000      56147 43352 73710 54211      11451 73020 20136 20230
   310      04400 00316 15623 20222      03060 00316 56371 20303      03330 00316 73220 37002      36770 51600 00127 46000
   314      01000 00531 61000 46000      56147 43352 73710 20136      43201 20260 12612 43352      20636 73200 54610 15613
   320      51600 00117 01000 00531      56171 20103 03210 00327      56147 43352 54211 15713      20230 21160 04400 00327
   324      36771 15623 20222 43353      03060 00327 15773 73220      51600 00127 01000 00531      04450 00276 21421 63440
   330      04000 00306 61000 46000      04000 00165 00000 00000      07100 00334 61117 77776      04100 00401 61110 00001
   334      07100 00336 61000 46000      01000 00335 00000 00001      66111 07100 00340 46000      01000 00337 00000 00001
   340      66111 07100 00342 46000      01000 00341 00000 00001      66111 07100 00344 46000      01000 00343 00000 00001
   344      66111 07100 00346 46000      01000 00345 00000 00001      66111 07100 00350 46000      01000 00347 00000 00001
   350      66111 07100 00352 46000      01000 00351 00000 00001      66111 07100 00354 46000      01000 00353 00000 00001
   354      66111 07100 00356 46000      01000 00355 00000 00001      66111 07100 00360 46000      01000 00357 00000 00001
   360      66111 07100 00362 46000      01000 00361 00000 00001      66111 07100 00364 46000      01000 00363 00000 00001
   364      66111 07100 00366 46000      01000 00365 00000 00001      66111 07100 00370 46000      01000 00367 00000 00001
   370      66111 07100 00372 46000      01000 00371 00000 00001      66111 07100 00374 46000      01000 00373 00000 00001
   374      66111 07100 00376 46000      01000 00375 00000 00001      66111 07100 00400 46000      01000 00377 00000 00001
   400      04000 00402 01000 00401      01000 00401 04000 00402      65170 51700 00155 10766      77610 61100 00001 46000
   404      51700 00154 10755 55771      10744 55771 10733 55771      10722 55771 10711 55771      10700 55771 43452 76770
   410      15664 15774 20622 12767      51700 00145 75660 77760      15664 15774 20622 12767      55771 75650 77750 15664
   414      15774 20622 12767 55771      75640 77740 15664 15774      20622 12767 55771 75630      77730 15664 15774 20622
   420      12767 55771 75620 77720      15664 15774 20622 12767      55771 75610 77710 15664      15774 20622 12767 55771
   424      51100 00401 20103 46000      03210 00430 55111 10711      20736 54711 04000 00437      01000 00333 00000 00001
   430      66210 61300 00044 66422      76120 51200 00427 20137      10622 76700 61000 46000      51320 00334 66224 20701
```

```
440     72617 77776 15774 53160     20622 12667 20622 46000     51600 00136 04000 00331     04000 00170 00000 00000

444     63260 53167 73610 46000     51600 00132 51600 00134     51100 00131 63310 66420     76400 43514 20536 46000
450     66540 56174 11651 46000     03060 00467 37765 46000     03070 00461 21621 63660     61447 77775 04460 00457
454     56174 73110 03210 00453     10455 01000 00566 46000     04000 00453 61000 46000     56176 73110 03210 00450
460     10455 04000 00467 46000     03140 00467 61000 46000     61447 77775 56174 73110     03210 00465 10455 46000
464     01000 00566 61000 46000     56174 11251 03020 00467     13724 03170 00462 46000     66420 46000 61000 46000
470     61440 00002 04340 00443     56174 11251 03020 00443     37224 03020 00443 73610     03260 00470 01000 00566
474     10455 04000 00470 46000     04000 00563 00000 00000     51600 00512 54761 10633     22704 54671 54761 76640
500     76750 54671 54761 76660     76770 54671 54761 46000     51100 00132 51200 00134     04000 00504 61000 46000
504     51100 00512 61000 46000     01000 11112 00000 00000     51100 00513 54311 54431     10711 54141 54211 63410
510     63520 54121 54211 63610     63720 04000 00475 46000     00000 00000 00000 11353     00000 00000 00000 11333
514     00000 00000 00000 11310     00000 00000 77770 00000     00000 00000 00000 00014     00000 00000 00000 00014
520     00000 00000 00000 00010     00000 00000 00000 00670     56135 54211 10611 22702     56625 61550 00002 54761
524     07540 00522 10644 22705     56670 36663 56771 36773     61770 00002 07760 00525     04000 03351 00000 00000
530     00000 00000 00004 00527     04000 00321 00000 00000     36127 50267 77772 43652     54721 11262 54771 73311
534     12772 54171 54720 11161     12631 54610 75271 46000     71707 77757 01000 00625     04000 00531 61000 46000
540     04000 00306 00000 00000     51100 00100 51200 00065     21122 73220 03320 00560     73110 03010 00564 37212
544     03320 00540 61000 46000     51100 00133 54610 37116     03210 00540 51200 00135     03020 00540 72660 00103
550     43166 21236 11616 37226     03020 00540 53060 20636     51600 00135 71601 50515     27606 20652 71100 00135
554     12661 01000 00633 46000     51200 00115 43652 11262     74600 12662 54620 46000     04000 00540 61000 46000
560     51100 00104 73110 37216     03320 00562 10611 46000     51600 00133 01000 00475     04000 00540 61000 46000
564     51100 00104 73110 37212     03020 00545 04000 00540     04000 00456 00000 00000     56147 43773 20721 11771
570     54710 54211 73620 73170     51300 00132 37731 21774     11373 15717 12773 54730     51300 00134 36716 37173
574     21174 15671 11313 12636     54630 20252 73320 56171     03030 00566 20103 37673     03210 00601 55121 73610
600     21160 36661 36763 46000     51300 00122 43152 11313     12663 54630 54761 73660     54671 54361 73771 11313
604     12773 20252 54730 15721     03170 00613 71600 00616     51600 00127 71200 00121     71707 77753 01000 00625
610     43652 51100 00616 15616     20644 54220 12626 54620     04000 00566 61000 46000     51700 00127 71200 00121
614     71707 77553 01000 00625     04000 00566 61000 46000     00000 00000 00000 00001     01000 00652 61000 46000
620     43652 71100 00002 12661     53120 11161 71600 31117     03270 00623 14777 27606     12717 20652 53720 12662
624     73220 01000 00633 46000     04000 00537 00000 00000     53120 20173 03310 00620     03110 00617 53121 73610
630     54611 54661 02000 00625     04000 00644 00000 00000     01300 00000 00000 00000     04000 00175 00000 00000
634     51100 00001 03110 00634     54610 04000 00632 46000     51100 00066 03310 00640     51100 00631 04000 00641
640     71100 00130 20160 46000     13661 13161 13661 46000     51600 00632 10611 46000     51100 00001 01000 00631
644     20652 01000 00633 46000     51100 00001 03110 00645     04004 00646 61000 46000     51100 00001 03110 00646
650     71602 20314 04000 00644     20150 36661 01000 00633     04004 00652 61000 46000     71602 20314 20652 36662
654     53160 20173 03310 00652     03010 00652 51100 00001     03110 00654 71100 00001     04000 00651 61000 46000
660     71603 24616 12661 20651     01000 00633 61000 46000     04004 00662 61000 46000     20630 12161 73610 20123
664     03210 00660 20151 13116     20636 51600 00667 74660     36116 20123 04000 00660     00000 00000 00000 00000
670     00050 00002 00004 01020     34011 35300 00000 10141     00030 00104 00000 11161     00000 00000 00000 00063
674     00000 00105 00010 11244     00000 00000 00000 00014     00000 00106 00010 11244     00000 00000 00000 00014
700     00030 00107 77774 11161     00000 00100 00010 00127     00000 00111 00040 11310     00000 00000 00000 00023
704     00000 00112 00044 11310     00000 00000 00000 00014     00000 00113 00004 11333     20000 00000 00000 00020
710     60000 00000 04004 00710     60000 00000 04004 00711     60000 00000 04004 00712     60000 00000 04004 00713
714     60000 00000 04004 00714     60000 00000 04004 00715     60000 00000 04004 00716     60000 00000 04004 00717
720     60000 00000 04004 00720     60000 00000 04004 00721     60000 00000 04004 00722     60000 00000 04004 00723
724     60000 00000 04004 00724     60000 00000 04004 00725     60000 00000 04004 00726     60000 00000 04004 00727
730     60000 00000 04004 00730     60000 00000 04004 00731     60000 00000 04004 00732     60000 00000 04004 00733
734     60000 00000 04004 00734     60000 00000 04004 00735     60000 00000 04004 00736     60000 00000 04004 00737
740     60000 00000 04004 00740     60000 00000 04004 00741     60000 00000 04004 00742     60000 00000 04004 00743
744     60000 00000 04004 00744     60000 00000 04004 00745     60000 00000 04004 00746     60000 00000 04004 00747
750     60000 00000 04004 00750     60000 00000 04004 00751     60000 00000 04004 00752     60000 00000 04004 00753
754     60000 00000 04004 00754     60000 00000 04004 00755     60000 00000 04004 00756     60000 00000 04004 00757
760     60000 00000 04004 00760     60000 00000 04004 00761     60000 00000 04004 00762     60000 00000 04004 00763
764     60000 00000 04004 00764     60000 00000 04004 00765     60000 00000 04004 00766     60000 00000 04004 00767
770     60000 00000 04004 00770     60000 00000 04004 00771     60000 00000 04004 00772     60000 00000 04004 00773
774     60000 00000 04004 00774     60000 00000 04004 00775     60000 00000 04004 00776     60000 00000 04004 00777
1000    60000 00000 04004 01000     60000 00000 04004 01001     60000 00000 04004 01002     60000 00000 04004 01003
1004    60000 00000 04004 01004     60000 00000 04004 01005     60000 00000 04004 01006     60000 00000 04004 01007
1010    60000 00000 04004 01010     60000 00000 04004 01011     60000 00000 04004 01012     60000 00000 04004 01013
1014    60000 00000 04004 01014     60000 00000 04004 01015     60000 00000 04004 01016     60000 00000 04004 01017
1020    01000 11165 00130 03350     00000 00000 00040 03356     00000 00000 00040 03365     00000 00000 00044 03373
1024    00000 00000 00074 03362     55555 55555 55555 55555     40404 04040 40404 04040     11162 02524 00000 00000
```

```
1034      00000 00000 00000 03344      00000 00000 00000 11610      00000 00000 00000 00000      00000 00000 00000 00000
1040      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00024

1044      00000 00000 00000 00000      00000 00000 00000 07640      20000 00000 00000 00106      00000 00000 00000 00000
1050      00000 00000 00000 01051      00000 00000 00000 00000      00000 00000 00000 00000      00000 00000 00000 00000
1054      04000 03376 00000 00000      55642 21717 24551 12355      02010 31355 01160 45524      05221 51116 01241 11607
1060      64000 00000 00000 00000      16045 55555 55645 56403      01141 46400 00000 00000      55555 55555 55555 55555
1064      55555 55555 55555 55555      55555 55555 55555 55555      55555 55555 55555 55555      55555 55555 55555 55555
      DUPLICATED LINES.
1074      00000 00000 00000 03427      00000 00000 67660 00000      10000 00000 00000 01265      00000 00000 00000 03402
1100      00060 00000 34000 0345E      00000 00000 00000 00000      20000 06162 40000 00001      00000 00000 00000 00000
```

```
CATALOG OF  DRF2755          FM/SYS172  81/10/07. 11.17.39.  PAGE   1   2

INDIRECT ACCESS FILE(S)

 BUILD      M9315      M9354      M9452      M9551      M9652      M9751
 DEMO       M9316      M9355      M9471      M9552      M9671      M9752
 MTS9J08    M9333      M9356      M9472      M9571      M9672      PAH
 M9000      M9334      M9357      M9501      M9591      M9701      RUN
 M9001      M9335      M9401      M9502      M9601      M9702      STLF
 M9286      M9336      M9402      M9521      M9602      M9721      TEST
 M9313      M9353      M9451      M9522      M9651      M9722      ZZZ
 M9314

DIRECT ACCESS FILE(S)

 QACQACR    STL

      50 INDIRECT ACCESS FILE(S),   TOTAL PRUS =       661.
       2 DIRECT ACCESS FILE(S),     TOTAL PRUS =       298.
```

# EXERCISE 13

PROBLEM

1. Access the source library and extract program MAIN, compile and, before executing, declare the 2nd user library built in exercise 11 (STEP 3) to be used by the loader to satisfy external references in MAIN.

2. Access two user libraries (created in exercise 11) both of which contain SUB1 and SUB2. Access the routine SUB1 and SUB2 in the following combinations:

| LIB1 | LIB2 |
|---|---|
| ① SUB1 | ② SUB1 |
| ② SUB2 | ① SUB2 |

① SUB1 from LIB1, SUB2 from LIB2

② SUB1 from LIB2, SUB2 from LIB1

# LESSON 9
# NON-ALLOCATABLE RESOURCES - MAGNETIC TAPE

LESSON PREVIEW:

In this lesson, the student will learn how to explicitly request one or more magnetic tapes and private disk packs. The various types of tapes available under NOS are examined, and actual requests for tapes and packs are demonstrated.

OBJECTIVES:

After completing this lesson the student should be able to -

- Describe the components and their operation in the non-allocatable resource management software.
- List the types of tape formatting available and describe the structure of each.
- State the differences between binary and coded mode operations.
- Use the VSN, REQUEST, LABEL, and LISTLB control statements to create and access tapes.

TRAINING AIDS:

Visuals V9-1 through V9-11.

PROJECTS:

Exercises 14 (study questions) and 15 (problem).

REFERENCES:

NOS Version 1 Reference Manual, Section 10, Appendix G.

## RESOURCE MANAGEMENT

Resources other than use of the central processing unit, central memory and mass storage by the job can also consist of magnetic tapes and auxiliary disk packs.

Magnetic tapes are requested by means of an ASSIGN, REQUEST, LABEL, or VSN control statements.

Public and private disk packs are requested by means of the PN parameter on a permanent file control statement.

The number of resources a user may access are stated in the user number's validation file. For any number of resources more than one, a RESOURC control statement must be issued by the job stating the qualities and maximum quantities required at any point in the job. This enables the system to utilize deadlock prevention routines which roll out jobs that have a potential of exceeding the number of available units.

There are techniques for requesting magnetic tapes either automatically or manually. It is suggested that automatic assignment procedures be used whenever possible, since it enhances the overall efficiency of the system.

Visual 9-1 shows the major components of the system involved with resource management.

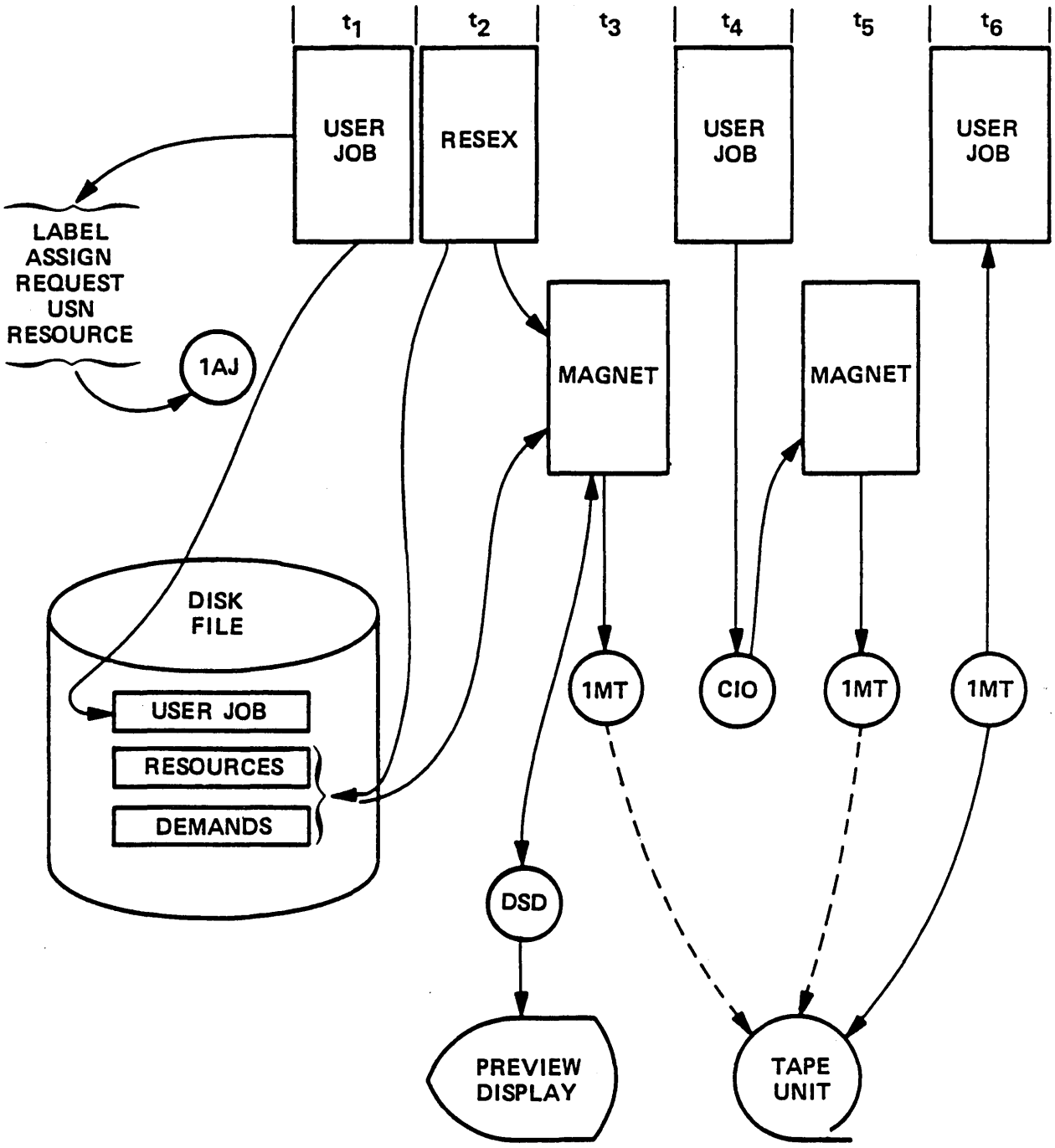
During the first time period $t_1$, the user job issues a resource request via an internal macro call and the advance job routine rolls the user job out before loading RESEX.

At $t_2$ - RESEX processes the request by making appropriate entries to resource and demand inventory files and passing on the request to MAGNET. RESEX then drops out and the control point is available for other jobs. If the user had issued a control card request, this would be the first time period.

At $t_3$ - MAGNET calls 1MT to actually locate the unit or make a request to the operator to locate and make ready a specific tape for assignment to the job via a "preview" display at the control console.

At $t_4$ - the user job, reactivated because the resource was assigned, now issues an operation on the tape and CIO (combined input/output) in a pool PPU processes the request and passes it along to MAGNET.

At $t_5$ - MAGNET further processes the request and passes it along to 1MT in a pool PPU, which selects the proper physical input/output driver routine and begins a transfer data during time period $t_6$.




Tape Request and Usage - Overview

## AUTOMATIC TAPE ASSIGNMENT

NOS processes tape requests as follows:

1. Whenever a tape is mounted, NOS checks for labels. If the tape was labeled, NOS keeps a record of the VSN read from VOL1 and the equipment on which the tape is mounted.

2. If, when a request is made for tape assignment, an LFN/VSN association is encountered, NOS compares the VSN associated with the file (or one of its equivalences) with the VSNs read from mounted tapes. If a match is found, the system automatically assigns the tape to the requesting job, provided a deadlock would not occur. If the tape is not mounted, NOS rolls out the job until a tape with the required VSN is mounted. For a mounted, unlabeled tape, the operator enters a VSN command specifying the required VSN. NOS is then able to automatically assign the tape.

3. If no LFN/VSN association is encountered when the request is made, NOS directs the operator to assign an available unit.

4. For an ASSIGN card, the method of assignment depends on the nn parameter. If nn is a device type (MT or NT), the operator must assign an available unit. If nn is the EST ordinal of a tape unit, NOS automatically assigns the specified unit.

The following is a summary of the system and/or operator action taken in response to an ASSIGN, LABEL, or REQUEST card. The VSN column indicates whether or not the user has specified an LFN/VSN association via the VSN keyword or a VSN card. The mode column shows the mode as determined by NOS in checking for labels.

| Card | VSN | Mode | Action |
|---|---|---|---|
| ASSIGN | Yes | Labeled | If the nn parameter is MT or NT, the operator must assign an available unit. If the nn parameter is the EST ordinal of a tape unit, assignment is automatic. |
| | Yes | Unlabeled | Same as when the VSN is specified and the tape is labeled. |
| | No | Labeled | Same as when the VSN is specified and the tape is labeled. |
| | No | Unlabeled | Same as when the VSN is specified and the tape is labeled. |

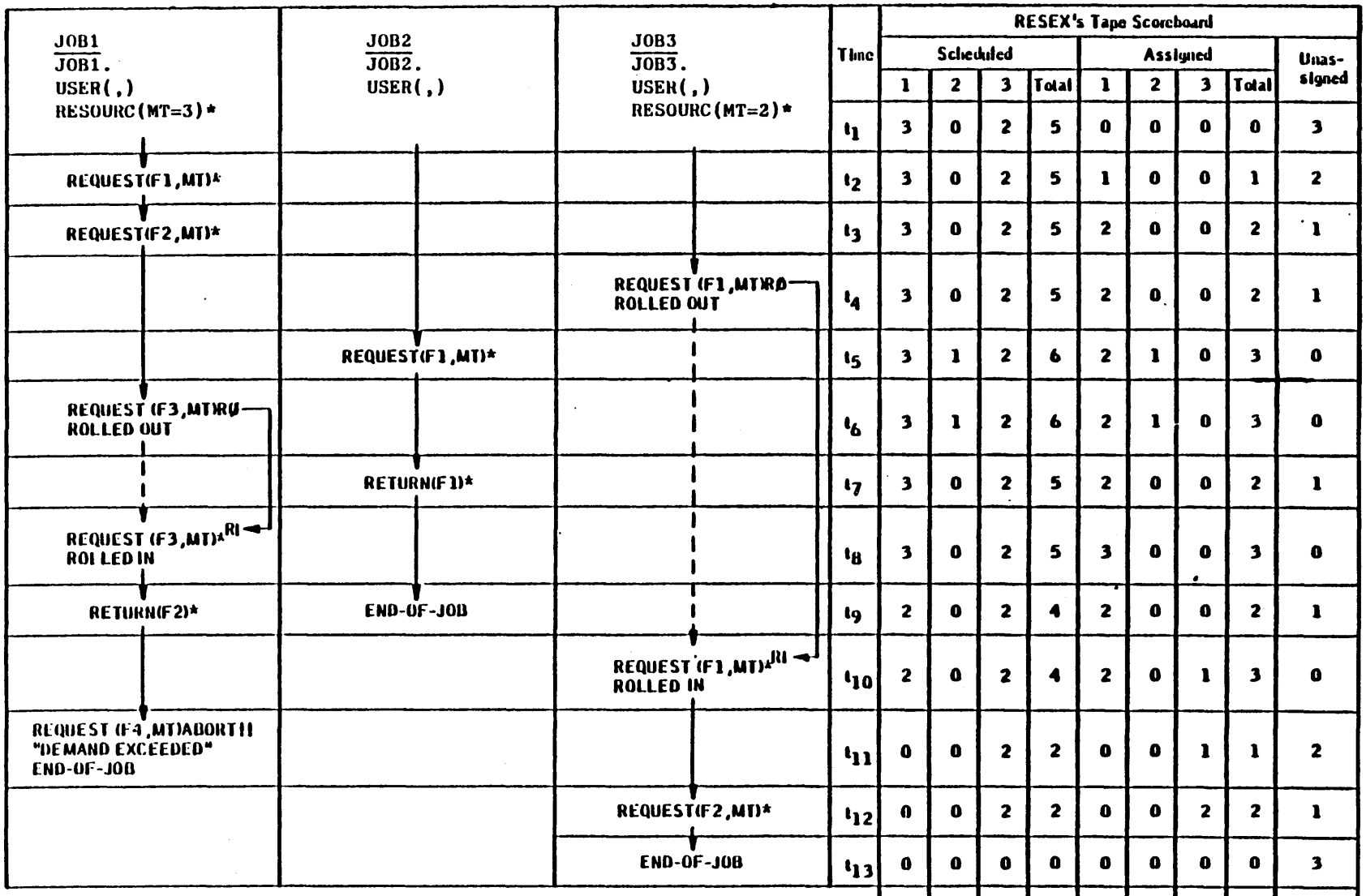
| Card | VSN | Mode | Action |
|---|---|---|---|
| REQUEST | Yes | Labeled | The system matches the VSN read from VOL1 with the VSN on the REQUEST or VSN card. Assignment is automatic. |
| | Yes | Unlabeled | The operator enters a VSN command specifying the VSN included on the REQUEST or VSN card. Assignment is automatic. |
| | No | Labeled | The operator assigns an available unit to LFN. |
| | No | Unlabeled | The operator assigned an available unit to LFN. |
| LABEL | Yes | Labeled | The system matches the VSN read from VOL1 with the VSN on the LABEL or VSN card. Assignment is automatic. |
| | Yes | Unlabeled | The operator enters a VSN command specifying the VSN included on the LABEL or VSN card. Assignment is automatic. |
| | No | Labeled | The operator assigns an available unit to LFN. |
| | No | Unlabeled | The operator assigns an available unit to LFN. |

Note that the LB keyword is not used in assigning a tape. Rather, it is used in processing the data on the tape once the assignment has been made.

## RESOURC CONTROL STATEMENT AND DEADLOCK PREVENTION

The discussion of the RESOURC control statement in section 6 of the NOS Reference Manual should be studied to understand resource control from a system viewpoint and how it is used by the programmer making a request for more than one non-allocatable device.

Visual V9-2 is a sequence of events in which a set of three tape units are in demand by three jobs and which want a total of six tape units at one point (compare $t_5$ and $t_6$).




| Time | RESEX's Tape Scoreboard | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Scheduled | | | | Assigned | | | | Unassigned |
| | 1 | 2 | 3 | Total | 1 | 2 | 3 | Total | |
| $t_1$ | 3 | 0 | 2 | 5 | 0 | 0 | 0 | 0 | 3 |
| $t_2$ | 3 | 0 | 2 | 5 | 1 | 0 | 0 | 1 | 2 |
| $t_3$ | 3 | 0 | 2 | 5 | 2 | 0 | 0 | 2 | 1 |
| $t_4$ | 3 | 0 | 2 | 5 | 2 | 0 | 0 | 2 | 1 |
| $t_5$ | 3 | 1 | 2 | 6 | 2 | 1 | 0 | 3 | 0 |
| $t_6$ | 3 | 1 | 2 | 6 | 2 | 1 | 0 | 3 | 0 |
| $t_7$ | 3 | 0 | 2 | 5 | 2 | 0 | 0 | 2 | 1 |
| $t_8$ | 3 | 0 | 2 | 5 | 3 | 0 | 0 | 3 | 0 |
| $t_9$ | 2 | 0 | 2 | 4 | 2 | 0 | 0 | 2 | 1 |
| $t_{10}$ | 2 | 0 | 2 | 4 | 2 | 0 | 1 | 3 | 0 |
| $t_{11}$ | 0 | 0 | 2 | 2 | 0 | 0 | 1 | 1 | 2 |
| $t_{12}$ | 0 | 0 | 2 | 2 | 0 | 0 | 2 | 2 | 1 |
| $t_{13}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |

*Request processed when encountered

RESOURC Deadlock Prevention

## TAPE RESOURCE TYPES - I, SI, S, L, AND F

Of the five types of tape available, the I (Internal) format is the default tape for the NOS. The tape being written in this mode is assumed to be read in the same mode. The other tape formats are provided for compatibility with tapes created by other operating systems or by predecessor systems to NOS.

The tape formats are discussed in section 10 of the NOS Reference Manual and should be studied for details of each format. Visual V9-3 summarizes the general characteristics of the five formats.

## RECORDING MODES - CODED OR BINARY

The recording mode of a tape refers to the physical manner in which the tape is written. Specifically, coded mode is written in even parity and binary mode in odd parity. The why of two-recording modes is implicit in the evolution of magnetic tape technology. The important thing to remember is that a tape created in one mode must be read in the same mode.

Binary mode is the only mode offered for I-tapes and this mode is recommended generally because its physical blocking is larger due to freedom from required code conversions. Binary mode is also the innate mode of the rest of the system.

## LABELED TAPES AND VSN

The VSN is the main field in providing a labeled tape library capability to an installation. The VSN field is either recorded physically on a VOL1 header label or assigned logically by the operator to an unlabeled tape. In any case, the VSN is the physical number that tells the tape librarian which reel is required for mounting onto a tape unit.

The VSN field in the VOL1 header label does not have to be the same as the number for operator's visual identification. It is not unusual for a tape to contain dissimilarities between internal and external VSNs. There is no confusion, because the VSN command allows equating two or more internal/external numbers.

The creation of a labeled tape is most easily done by use of the LABEL and control statement. There are other, more elaborate methods of creating labels which involve using macros and setting fields in the File Environment Table.

The label formats of the ANSI standard labels are in appendix G of the NOS Reference Manual.

| TYPE | MODE | BLOCK SIZE | EOR | EOF | EOI | COMMENTS |
|---|---|---|---|---|---|---|
| I | Binary | $0 \leqslant BS \leqslant 512$ CMWDS | $BS < 512$ CMWDS + Code word level zero | $BS = 0$ + code word level $17_8$ | Labeled-TM<br>---<br>Unlabeled none | ---<br>Most reliable of all formats |
| SI | Binary<br>---<br>Coded | $0 \leqslant BS \leqslant 512$ CMWDS<br>---<br>$0 \leqslant BS \leqslant 128$ CMWDS | $BS < 512$ CMWD + level $0 \longrightarrow 16_8$<br>---<br>$BS < 128$ CMWD + level $0 \longrightarrow 16_8$ | $BS = 0$ + code of level $17_8$ | TM + EOF1 trailer label | Standard with NOS/BE operating systems |
| S | Binary Coded | $0 \leqslant BS \leqslant 512$ CMWDS | The data between IRG's | | None | A variable record length tape from NOS/BE |
| L | Binary Coded | $0 \leqslant BS \leqslant$ CM Buffer Length | The data between IRG's | None | None | A variable record length tape from NOS/BE |
| F | Binary Coded | $0 \leqslant BS \leqslant$ CM Buffer Length | The data between IRG's | TM | None | Foreign |

NOS Tape Formats, Summary

```
MTJOB1.
USER(JCL4073,JCL4)
CHARGE(6881,77143408)
REQUEST,XYZ,MT,F=I,LB=KL,D=800,PO=U.     ASSIGN ANY SCRATCH TAPE
LABEL(XYZ,VSN=000PL3,MT,F=I,D=800,PO=U)
COPY,,XYZ.
REWIND,XYZ.
TDUMP(I=XYZ)
000000000000000000000     7/8/9 END-OF-RECORD
 THIS IS RECORD 1 LINE 1
 THIS IS RECORD 1 LINE 2
 THIS IS RECORD 1 LINE 3
 THIS IS RECORD 1 LINE 4
000000000000000000000     7/8/9 END-OF-RECORD
 THIS IS RECORD 2 LINE 1
 THIS IS RECORD 2 LINE 2
 THIS IS RECORD 2 LINE 3
 THIS IS RECORD 2 LINE 4
000000000000000000000     6/7/8/9 END-OF-INFORMATION
```

```
AQOYESS. 79/07/10. (35) CYBER 174 S/N 620 CLSH.
```

```
19.34.44.MTJOB1.
19.34.44.UCCR, L1CT1  ,       0.019KCDS.
19.34.44.USER(JCL4073,)
19.34.44.CHARGE(6881,77143408)
19.34.44.REQUEST,XYZ,MT,F=I,LB=KL,D=800,PO=U.
19.34.44.ASSIGN ANY SCRATCH TAPE
19.35.42.MT53, ASSIGNED TO XYZ     , VSN=EXAMPL.
19.35.42.LABEL(XYZ,VSN=000PL3,MT,F=I,D=800,PO=U)
19.35.43.COPY,,XYZ.
19.35.43. EOI ENCOUNTERED.
19.35.43.REWIND,XYZ.
19.35.43.TDUMP(I=XYZ)
19.35.46. TDUMP COMPLETE.
19.35.46.UEAD,        0.004KUNS.
19.35.46.UEPF,        0.021KUNS.
19.35.46.UEMT,        0.030KUNS.
19.35.46.UEMS,        0.664KUNS.
19.35.46.UECP,        0.095SECS.
19.35.46.AESR,        4.080UNTS.
19.36.41.UCLP, L1CT1  ,       0.125KLNS.
```

```
        - FILE DUMP -        TDUMP(I=XYZ)                                                    79/07/10. 19.35.43. PAGE    1.

F  1 R   1 W     0- 5524 1011 2355 1123 5522  0503 1722 0455 3455 1411  1605 5534 0000 0000 0000  5524 1011 2355 1123 5522
                      T  H I  S    I S    R   E C  O R  D    1    L I   N E    1                    T  H I  S    I S    R
F  1 R   1 W     4- 0503 1722 0455 3455 1411  1605 5535 0000 0000 0000  5524 1011 2355 1123 5522  0503 1722 0455 3455 1411
                   E C  O R  D    1    L I   N E    2                    T  H I  S    I S    R   E C  O R  D    1    L I
F  1 R   1 W    10- 1605 5536 0000 0000 0000  5524 1011 2355 1123 5522  0503 1722 0455 3455 1411  1605 5537 0000 0000 0000
                   N E    3                    T  H I  S    I S    R   E C  O R  D    1    L I   N E    4
-- END OF RECORD --

F  1 R   2 W     0- 5524 1011 2355 1123 5522  0503 1722 0455 3555 1411  1605 5534 0000 0000 0000  5524 1011 2355 1123 5522
                      T  H I  S    I S    R   E C  O R  D    2    L I   N E    1                    T  H I  S    I S    R
F  1 R   2 W     4- 0503 1722 0455 3555 1411  1605 5535 0000 0000 0000  5524 1011 2355 1123 5522  0503 1722 0455 3555 1411
                   E C  O R  D    2    L I   N E    2                    T  H I  S    I S    R   E C  O R  D    2    L I
F  1 R   2 W    10- 1605 5536 0000 0000 0000  5524 1011 2355 1123 5522  0503 1722 0455 3555 1411  1605 5537 0000 0000 0000
                   N E    3                    T  H I  S    I S    R   E C  O R  D    2    L I   N E    4
-- END OF RECORD --

-- END OF FILE --

-- END OF INFORMATION --

-- END OF DUMP --
```

```
MTJOB2,T5.
USER(JCL4073,JCL4)
CHARGE(6881,77143408)
VSN(XYZ=EXAMPL)
LABEL(XYZ,F=I,FI=CLASSEXAMPLE,W,MT,PO=U,D=800)
REWIND(XYZ)
TDUMP(I=XYZ)
RETURN(XYZ)
VSN(XYZ=EXAMPL)
LABEL,XYZ,R,FI=CLASSKXAMPLE,MT,F=I,PO=U,D=800.
EXIT.
00000000000000000000    6/7/8/9 END-OF-INFORMATION
```

```
        - FILE DUMP -        TDUMP(I=XYZ)

-- END OF INFORMATION --

-- END OF DUMP --
```

```
AQOYETA. 79/07/10. (35) CYBER 174 S/N 620 CLSH.
```

```
19.37.16.MTJOB2,T5.
19.37.16.UCCR, L1CT1  ,       0.012KCDS.
19.37.16.USER(JCL4073,)
19.37.16.CHARGE(6881,77143408)
19.37.16.VSN(XYZ=EXAMPL)
19.37.17.LABEL(XYZ,F=I,FI=CLASSEXAMPLE,W,MT,PO=U,
19.37.17.D=800)
19.37.17.MT53, ASSIGNED TO XYZ     , VSN=EXAMPL.
19.37.17.REWIND(XYZ)
19.37.17.TDUMP(I=XYZ)
19.37.22. TDUMP COMPLETE.
19.37.22.RETURN(XYZ)
19.37.22.VSN(XYZ=EXAMPL)
19.37.22.LABEL,XYZ,R,FI=CLASSKXAMPLE,MT,F=I,PO=U,
19.37.22.D=800.
19.37.27.MT53, ASSIGNED TO XYZ     , VSN=EXAMPL.
19.37.27. LABEL PARAMETER CONFLICT ON OPEN, XYZ A
19.37.27.T 120.
19.37.27. FIELD BEGINNING AT 05 NO COMPARE.
19.37.27.EXIT.
19.37.27.UEAD,       0.006KUNS.
19.37.27.UEPF,       0.061KUNS.
19.37.27.UEMT,       0.005KUNS.
19.37.27.UEMS,       1.458KUNS.
19.37.27.UECP,       0.135SECS.
19.37.27.AESR,       6.053UNTS.
19.38.07.UCLP, L1CT1  ,       0.114KLNS.
```

```
MTJOB3,T4.
USER(JCL4073,JCL4)
CHARGE(6881,77143408)
*****************************************
* WRITE A LABELLED TAPE                 *
*****************************************
BLANK(VSN=ABCD,D=800,MT,LSL=1)
LABEL,XYZ,F=I,FI=ISNTTHISFUN,D=800,W,MT,VSN=ABCD,PO=U.
COPYBR(INPUT,XYZ)
REWIND,XYZ.
COPY(XYZ,ABC,Z)
COPYSBF(ABC,OUTPUT)
0000000000000000000     7/8/9 END-OF-RECORD
 DATA CARD NUMBER ONE
 DATA CARD NUMBER TWO
 DATA CARD NUMBER THREE
 DATA CARD NUMBER FOUR
 DATA CARD NUMBER FIVE
0000000000000000000     6/7/8/9 END-OF-INFORMATION
```

```
 DATA CARD NUMBER ONE
 DATA CARD NUMBER TWO
 DATA CARD NUMBER THREE
 DATA CARD NUMBER FOUR
 DATA CARD NUMBER FIVE
```

```
AQOYETF. 79/07/10. (35) CYBER 174 S/N 620 CLSH.
```

```
19.38.42.MTJOB3,T4.
19.38.42.UCCR, L1CT1  ,       0.019KCDS.
19.38.42.USER(JCL4073,)
19.38.42.CHARGE(6881,77143408)
19.38.42.*****************************************
19.38.42.* WRITE A LABELLED TAPE                 *
19.38.42.*****************************************
19.38.42.BLANK(VSN=ABCD,D=800,MT,LSL=1)
19.39.28.MT53, ASSIGNED TO BLKTAPE, VSN=EXAMPL.
19.39.32. TAPE BLANK LABELED.
19.39.32.LABEL,XYZ,F=I,FI=ISNTTHISFUN,D=800,W,MT,
19.39.32.VSN=ABCD,PO=U.
19.39.37.MT53, ASSIGNED TO XYZ     , VSN=ABCD   .
19.39.37.COPYBR(INPUT,XYZ)
19.39.37. COPY COMPLETE.
19.39.37.REWIND,XYZ.
19.39.37.COPY(XYZ,ABC,Z)
19.39.42. EOI ENCOUNTERED.
19.39.42.*  VERIFY,XYZ,ABC,A,R,L=OUTPUT,N=0.
19.39.44. VERIFY GOOD.
19.39.44.COPYSBF(ABC,OUTPUT)
19.39.44. END OF INFORMATION ENCOUNTERED.
19.39.44.UEAD,        0.004KUNS.
19.39.44.UEPF,        0.045KUNS.
19.39.44.UEMT,        0.057KUNS.
19.39.44.UEMS,        1.279KUNS.
19.39.44.UECP,        0.172SECS.
19.39.44.AESR,        4.121UNTS.
19.41.05.UCLP, L1CT1  ,       0.166KLNS.
```

```
MTJOB4,T6.
USER(JCL4073,JCL4)
CHARGE(6881,77143408)
*****************************************
* WRITE A LABELLED TAPE                 *
*****************************************
LABEL(XYZ,F=I,FI=NOSISOKAY,D=800,W,MT,VSN=ABCD,PO=U)
COPYBR,,XYZ.
REWIND(XYZ)
COPY(XYZ,ABC,V)
COPYSBF(ABC)
00000000000000000000    7/8/9 END-OF-RECORD
 ANOTHER DATA CARD 1
 ANOTHER DATA CARD 2
 ANOTHER DATA CARD 3
 ANOTHER DATA CARD 4
 ANOTHER DATA CARD 5
00000000000000000000    6/7/8/9 END-OF-INFORMATION
```

```
ANOTHER DATA CARD 1
ANOTHER DATA CARD 2
ANOTHER DATA CARD 3
ANOTHER DATA CARD 4
ANOTHER DATA CARD 5
```

```
AQOYETL. 79/07/10. (35) CYBER 174 S/N 620 CLSH.



19.40.57.MTJOB4,T6.
19.40.57.UCC=, L1CT1  ,      0.018KCDS.
19.40.57.USER(JCL4073,)
19.40.57.CHARGE(6881,77143408)
19.40.57.*****************************************
19.40.57.* WRITE A LABELLED TAPE                 *
19.40.57.*****************************************
19.40.57.LABEL(XYZ,F=I,FI=NOSISOKAY,D=800,W,MT,VS
19.40.57.N=ABCD,PO=U)
19.40.58.MT53, ASSIGNED TO XYZ    , VSN=ABCD  .
19.40.58.COPYBR,,XYZ.
19.40.58. COPY COMPLETE.
19.40.58.REWIND(XYZ)
19.40.58.COPY(XYZ,ABC,V)
19.41.02. EOI ENCOUNTERED.
19.41.02.*  VERIFY,XYZ,ABC,A,R,L=OUTPUT,N=0.
19.41.04. VERIFY GOOD.
19.41.04.COPYSBF(ABC)
19.41.04. END OF INFORMATION ENCOUNTERED.
19.41.04.UEAD,       0.003KUNS.
19.41.04.UEPF,       0.029KUNS.
19.41.04.UEMT,       0.017KUNS.
19.41.04.UEMS,       0.762KUNS.
19.41.04.UECP,       0.098SECS.
19.41.04.AESR,       3.082UNTS.
19.41.57.UCLP, L1CT1  ,      0.160KLNS.
```

```
MTJOB5,T7.
USER(JCL4073,JCL4)
CHARGE(6881,77143408)
*****************************************
* WRITE AN UNLABELLED TAPE              *
*****************************************
REQUEST,XYZ,MT,F=I,LB=KU,D=800,PO=U.
COPYBR,,XYZ.
REWIND(XYZ)
COPY(XYZ,ABC)
REWIND(ABC)
COPYSBF(ABC)
00000000000000000000    7/8/9 END-OF-RECORD
 THIS IS ABSOLUTELY
 THE LAST DATA CARD
 I AM GOING TO PUNCH
 TONIGHT - GOODNIGHT
00000000000000000000    6/7/8/9 END-OF-INFORMATION
```

```
THIS IS ABSOLUTELY
THE LAST DATA CARD
I AM GOING TO PUNCH
TONIGHT - GOODNIGHT
```

```
AQOYETQ. 79/07/10. (35) CYBER 174 S/N 620 CLSH.



19.41.56.MTJOB5,T7.
19.41.56.UCCR, L1CT1  ,      0.018KCDS.
19.41.56.USER(JCL4073,)
19.41.56.CHARGE(6881,77143408)
19.41.56.*****************************************
19.41.56.* WRITE AN UNLABELLED TAPE              *
19.41.56.*****************************************
19.41.56.REQUEST,XYZ,MT,F=I,LB=KU,D=800,PO=U.
19.42.05.MT53, ASSIGNED TO XYZ    , VSN=ABCD  .
19.42.05.COPYBR,,XYZ.
19.42.06. COPY COMPLETE.
19.42.06.REWIND(XYZ)
19.42.06.COPY(XYZ,ABC)
19.42.08. EOI ENCOUNTERED.
19.42.08.REWIND(ABC)
19.42.08.COPYSBF(ABC)
19.42.08. END OF INFORMATION ENCOUNTERED.
19.42.08.UEAD,       0.003KUNS.
19.42.08.UEPF,       0.021KUNS.
19.42.08.UEMT,       0.008KUNS.
19.42.08.UEMS,       0.663KUNS.
19.42.08.UECP,       0.070SECS.
19.42.08.AESR,       3.056UNTS.
19.43.13.UCLP, L1CT1  ,      0.154KLNS.
```

```
MTJOB6,T10.
USER(JCL4073,JCL4)
CHARGE(6881,77143408)
COPYBR,,TAPE1.
FTN,B=STATREP,L=0.
*****************************************
* CREATE FIRST FILE ON TAPE - PROCEDURE*TO RUN JOB
*****************************************
LABEL(TAPE9,VSN=ABCD,F=I,QN=1,SI=ACCT,FI=PROCEDURE,LB=KL,W,PO=U)
COPYBR,,TAPE9.
*****************************************
* CREATE SECOND FILE ON TAPE            *
*****************************************
LABEL,TAPE9,VSN=ABCD,F=I,SI=ACCT,FI=STATUS REP PROG,LB=KL,QN=9999.
REWIND(STATREP)
COPYBF,STATREP,TAPE9.
*****************************************
* CREATE THIRD FILE ON TAPE DATA FOR   * REPORT
*****************************************
LABEL(TAPE9,VSN=ABCD,F=I,SI=ACCT,FI=ACCT DATA,LB=KL,QN=9999.
COPYBR,TAPE1,TAPE9.
RETURN(TAPE9)
LABEL(TAPE9,MT,VSN=ABCD,PO=U,SI=ACCT)
LISTLB(TAPE9,SI=ACCT,LO=A)
RETURN(TAPE9)
7/8/9 END-OF-RECORD
 REPLACE THIS CARD WITH DATA FOR JOB
7/8/9 END-OF-RECORD
      PROGRAM TEST(OUTPUT)
      PRINT *,*THIS IS A DUMMY PROGRAM*
      END
7/8/9 END-OF-RECORD
 REPLACE THIS CARD WITH PROCEDURE TO RUN JOB
REQUEST(TAPE9,MT,VSN=ABCD,F=S,PO=U,LB=KU)
TDUMP(I=TAPE9,R=2)
REQUEST(TAPE9,MT,VSN=ABCD,PO=U)
LISTLB(TAPE9,LO=R)
6/7/8/9 END-OF-INFORMATION
```

```
AQOYCYM. 79/07/11. (35) CYBER 174 S/N 620 CLSH.

15.49.19.MTJOB6,T10.
15.49.19.UCCR. LICTI  .       0.039KCDS.
15.49.19.USER(JCL4073,)
15.49.19.CHARGE(6881.7714340H)
15.49.19.COPYBR,,TAPE1.
15.49.19. COPY COMPLETE.
15.49.19.FTN,B=STATREP,L=0.
15.49.21.       .042 CP SECONDS COMPILATION TIME
15.49.21.*****************************************
15.49.21.* CREATE FIRST FILE ON TAPE - PROCEDURE*
15.49.21.TO RUN JOB
15.49.21.*****************************************
15.49.21.LABEL(TAPE9,VSN=ABCD,F=I,QN=1,SI=ACCT,FI
15.49.21.=PROCEDURE,LB=KL,W,PO=U)
15.49.21.MT55. ASSIGNED TO TAPE9  . VSN=ABCD  .
15.49.22.COPYBR,,TAPE9.
15.49.22. COPY COMPLETE.
15.49.22.*****************************************
15.49.22.* CREATE SECOND FILE ON TAPE            *
15.49.22.*****************************************
15.49.22.LABEL,TAPE9,VSN=ABCD,F=I,SI=ACCT,FI=STAT
15.49.22.US REP PROG,LB=KL,QN=9999.
15.49.23.REWIND(STATREP)
15.49.23.COPYBF,STATREP,TAPE9.
15.49.23. COPY COMPLETE.
15.49.24.*****************************************
15.49.24.* CREATE THIRD FILE ON TAPE DATA FOR   *
15.49.24. REPORT
15.49.24.*****************************************
15.49.24.LABEL(TAPE9,VSN=ABCD,F=I,SI=ACCT,FI=ACCT
15.49.24. DATA,LB=KL,QN=9999.
15.49.24.COPYBR,TAPE1,TAPE9.
15.49.25. EOI ENCOUNTERED.
15.49.25.RETURN(TAPE9)
15.49.25.LABEL(TAPE9,MT,VSN=ABCD,PO=U,SI=ACCT)
15.49.30.MT55. ASSIGNED TO TAPE9  . VSN=ABCD  .
15.49.30.LISTLB(TAPE9,SI=ACCT,LO=A)
15.49.35. MULTI-FILE NOT FOUND. TAPE9 AT 110.
15.49.35. REQUESTED SECTION  0004.
15.49.35. FOUND SECTION  0003.
15.49.36. LISTLB COMPLETE.
15.49.36.RETURN(TAPE9)
15.49.36.UEAD.       0.006KUNS.
15.49.36.UEPF.       0.053KUNS.
15.49.36.UEMT.       0.043KUNS.
15.49.36.UEMS.       2.188KUNS.
15.49.36.UECP.       0.210SECS.
15.49.36.AESR.       6.283UNTS.
15.51.16.UCLP. LICTI  .       0.266KLNS.
```

```
VOL1 LABEL READ:    VOL1ABCD                              NOSCLSHJCL4073                    1
 VSN=ABCD  , VA= , OWNERID=NOSCLSHJCL4073, LSL=1.
HDR1 LABEL READ:    HDR1PROCEDURE         ACC:  00010001000100 79192 79192 000000NOS   1.4-55
 FI=PROCEDURE       , SI=ACCT  , SN=0001, QN=0001, G=0001, E=00, CR= 79192, RT= 79192, FA= .
EOF1 LABEL READ:    EOF1PROCEDURE         ACC:  00010001000100 79192 79192 000001NOS   1.4-55
 FI=PROCEDURE       , SI=ACCT  , SN=0001, QN=0001, G=0001, E=00, CR= 79192, RT= 79192, FA= .
HDR1 LABEL READ:    HDR1STATUSREPPROG     ACC:  00010002000100 79192 79192 000000NOS   1.4-55
 FI=STATUSREPPROG   , SI=ACCT  , SN=0001, QN=0002, G=0001, E=00, CR= 79192, RT= 79192, FA= .
EOF1 LABEL READ:    EOF1STATUSREPPROG     ACC:  00010002000100 79192 79192 000002NOS   1.4-55
 FI=STATUSREPPROG   , SI=ACCT  , SN=0001, QN=0002, G=0001, E=00, CR= 79192, RT= 79192, FA= .
HDR1 LABEL READ:    HDR1ACCTDATA          ACC:  00010003000100 79192 79192 000000NOS   1.4-55
 FI=ACCTDATA        , SI=ACCT  , SN=0001, QN=0003, G=0001, E=00, CR= 79192, RT= 79192, FA= .
EOF1 LABEL READ:    EOF1ACCTDATA          ACC:  00010003000100 79192 79192 000000NOS   1.4-55
 FI=ACCTDATA        , SI=ACCT  , SN=0001, QN=0003, G=0001, E=00, CR= 79192, RT= 79192, FA= .
        7 LABELS READ.                7 LABELS PRINTED.
```

# EXERCISE 14

STUDY QUESTIONS

1. (true/false) All users have access to magnetic tapes.

2. (true/false) A RESOURC control card is required if more than one tape is to be used simultaneously by one job.

3. (true/false) A tape must be BLANK labeled before a real label can be written on it.

4. (true/false) A redundant request for a tape (one which is already assigned to the job) is ignored.

5. (true/false) Though it is valid to create multiple sets on both labeled and unlabeled tapes, only labeled tapes are positioned automatically to a given file.

6. (true/false) A REQUEST statement for an already mounted tape will provide automatic assignment if the VSN is specified for the file.

7. (true/false) Any user can issue an ASSIGN statement.

8. Explain the way in which the deadlock prevention function works in the NOS non-allocatable device subsystem.

# EXERCISE 15

PROBLEM

1. Request an NOS standard labeled scratch tape and write the FTN programs MAIN, SUB1, SUB2, and SUB3 on it.

2. Request the tape created in No. 1 above and compile and execute the programs on it.

3. Request a standard labeled scratch tape and write the FTN programs MAIN, SUB1, SUB2, and SUB3 on it as the first file of a multifile set and write the relocatable binaries of the FTN programs on it as the second file of the multifile set.

4. Request the tape created in No. 3 above and execute the relocatable binaries from the second file of the multifile set.

# LESSON 10

# FILE TRANSFERRING

THE ROUTE STATEMENT QUEUES A *TEMPORARY FILE FOR DISPOSAL TO THE INPUT QUEUE AS A NEW INDEPENDENT BATCH JOB OR TO THE OUTPUT QUEUE TO PRINT, PUNCH, PLOT, OR WAIT. THE ROUTED FILE CAN BE DISPOSED ON THE CURRENT MACHINE OR ANOTHER NOS MACHINE CONNECTED ON THE LCN (LOOSELY COUPLED NETWORK). THE FILE ROUTING TO THE OUTPUT QUEUE MAY TAKE EFFECT WHEN THE STATEMENT IS PROCESSED, A LATER JOB STEP, OR AT THE END OF JOB. TO ROUTE A FILE/JOB TO ANOTHER NON-NOS SYSTEM, THE MFLINK/MFQUEUE STATEMENT MUST BE USED.

EVERY JOB OR QUEUED FILE WILL HAVE A UNIQUE JSN (JOB SEQUENCE NUMBER) ASSOCIATED WITH IT. JSN WILL BE USED BY THE SYSTEM TO CONTROL JOBS.

*TEMPORARY FILE: THE FILE TO BE ROUTED MUST BE A LOCAL, NOT A DIRECT ACCESS PERMANENT FILE, AND NOT A PRIMARY FILE. AFTER FILE DISPOSAL, THE FILE IS NO LONGER LOCAL.

CREATING JOBS:

THE ROUTED FILE MUST BE IN A NORMAL JOB INPUT FORMAT HAVING A VALID JOBCARD AND USER CARD. TO ROUTE A JOB TO THE INPUT QUEUE (CREATE A BATCH JOB INDEPENDENT OF THE TERMINAL JOB) A DISPOSITION CODE (DC) OF EITHER IN, NO, OR TO MUST BE USED.

AT JOB TERMINATION IF:

IN: THE FILE OUTPUT ** WOULD BE PRINTED AT THE DEFAULT DEVICE FOR THE ROUTER, UNLESS EXPLICITLY ROUTED.

NO: THE FILE OUTPUT ** WOULD BE DISCARDED AT JOB TERMINATION UNLESS EXPLICITLY ROUTED.

TO: THE FILE OUTPUT ** WOULD BE PUT IN THE WAIT QUEUE FOR THE USER OWNING THE JOB DOING THE ROUTING UNLESS EXPLICITLY ROUTED.

THE PARAMETER ST ON THE ROUTE STATEMENT OR JOB COMMAND FOR INPUT QUEUE DISPOSITION WILL QUEUE THE JOB TO ANOTHER NOS SYSTEM IN THE LCN.

**OUTPUT FILE: THE OUTPUT FILE IS A UNIQUE LOCAL FILE PER JOB CALLED OUTPUT. IT IS A NON-ASCII FILE WHICH AT JOB TERMINATION WILL HAVE A COPY OF THE DAYFILE AS ITS LAST RECORD AND WHOSE JOB SEQUENCE NUMBER (JSN) WILL BE THE SAME AS THE JOB'S JSN.

Routing Output.

Using the route statement, any temporary file may be routed to any type of URD (Unit Record Device) output device by specifying a particular device by specifying either ID (Identifier id) or FC (Format Code). ST parameter can specify routing to another NOS system

Routing may occur immediately, or at job termination or later job step if DEF parameter specified.

The files OUTPUT, PUNCH, PUNCHB, and P8 are implicitly routed at job termination.

ST=L10 specifies the logical identifier of the remote host to which the file is to be routed.

SETJOB.

THE SETJOB COMMAND CHANGES THE USER JOB NAME (UJN), CAN REDIRECT THE DISPOSITION OF THE FILE OUTPUT ** IF THE JOB IS NOT INTERACTIVE OR BECOMES A DETACHED JOB, AND CAN CHANGE THE JOB PROCESSING OPTION AT JOB TERMINATION OR JOB SUSPENSION.

UJN IS USED WITH COMMANDS SUCH AS ENQUIRE, QGET, DROP, AND OTHERS WHERE JSN (JOB SEQUENCE NUMBER) COULD BE USED TO IDENTIFY JOBS OR QUEUED FILES THAT BELONG TO THE USER. JSN IS A SYSTEM VARIABLE WHILE UJN WOULD BE A KNOWN VALUE.

THE JOB PROCESSING OPTION SPECIFIES FOR DETACHED INTERACTIVE JOBS WHETHER THE JOB HAS TERMINATED (TJ), SUSPENDED (SU) FOR A SPECIFIED SYSTEM TIME WHEN AN ERROR OCCURS, HAS RUN OUT OF CONTROL CARDS, OR HAS REQUESTED I/O FROM TERMINAL.

EXAMPLE 1:

```
SETJOB, UJN=JOEY
ROUTE, MARY, DC=PR,REP=3.
   ROUTE COMPLETE.  JSN IS ACPN
```

WOULD ROUTE THE LOCAL FILE MARY TO THE PRINTER. THREE COPIES OF THE FILE WOULD BE PRINTED. ANY FUTURE REFERENCE TO DROP OR STATUS THE PRINT FILE COULD BE DONE BY USING JSN=ACPN OR UJN=JOEY.

NOTE: THE FILE MARY IS NO LONGER LOCAL.

EXAMPLE 2

```
ROUTE,CPUMTR,DC=NO.
ROUTE COMPLETE.  JSN IS ACPR
```

WOULD CREATE A BATCH JOB. AT JOB TERMINATION, NO OUTPUT FILE CONTAINING THE DAYFILE WOULD BE PRINTED.

EXAMPLE 3

```
ROUTE,JOHN,DC=PR,DEF.
ROUTE COMPLETE.
.
.
.
.
ROUTE,JOHN.

     ROUTE COMPLETE.   JSN IS ACRB

BYE.
```

OR

```
ROUTE, JOHN, DC = PR, DEF,
ROUTE COMPLETE.
.
.
.
.
BYE
```

BOTH SEQUENCES OF COMMANDS WILL CAUSE THE FILE JOHN TO BE PRINTED AT JOB TERMINATION.

A USER MAY ROUTE JOBS OR FILES TO OTHER NOS SYSTEMS USING THE ROUTE STATEMENT. TO ROUTE TEMPORARY FILES OR PERMANENT FILES TO OTHER NOS OR NON-NOS SYSTEMS THE STATEMENTS MFQUEUE AND MFLINK MUST BE USED.

MFLINK STATEMENT ALLOWS THE USER TO STORE FILE ON OR RETRIEVE PERMANENT FILES FROM A REMOTE HOST. THE USER MAY CHANGE THE SIZE OF THE FILE, CHANGE THE PASSWORD ASSOCIATED WITH A FILE, PURGE A FILE OR WHATEVER IS ALLOWED BY THE SERVICES ON THE REMOTE HOST. STAGING FILES TO ANOTHER NOS SYSTEM ALLOWS

| | |
|---|---|
| APPEND | SAVE |
| CHANGE | ATTACH |
| LET | DEFINE |
| PACKNAM | USER |
| PERMIT | CHARGE |
| REPLACE | PURGE |

FOR NON-NOS FILE STAGING, THE DIRECTIVES RESEMBLE (OR, IN FACT, ARE) COMMANDS FOR THAT OPERATING SYSTEM.

EXAMPLES: (NOS TO NOS)

A. ASSUME THE USER TO BE AT MAINFRAME MFB. TO TRANSFER A FILE FROM MAINFRAME MFA TO MFB AND PURGE IT ON MFA, THE USER MUST ENTER COMMANDS AND DIRECTIVES SIMILAR TO THE FOLLOWING, WHEN IN BATCH MODE:

```
MFLINK,NEWFILE,ST=MFA.
*ATTACH,OLDFILE/PN=USER.
*PURGE,OLDFILE/PN=USER.
```

B. ASSUME THE USER TO BE ON MAINFRAME MFB. TO TRANSFER FOUR FILES FROM MAINFRAME MFA, ON MFB ENTER COMMANDS AND DIRECTIVES SIMILAR TO THE FOLLOWING:

```
MFLINK,FILE1,ST=MFA,I=DIRECTS.
MFLINK,FILE2,I=DIRECTS.
MFLINK,FILE3,I=DIRECTS.
MFLINK,FILE4,I=DIRECTS.
```

WHERE THE CONTENTS OF FILE DIRECTS ARE:

```
USER,XYZ,ABC.
CHARGE,CHGN,PROJN.
GET,LFN1.
7/8/9
GET,LFN2.
7/8/9
ATTACH,PFN1/PN=USER.
7/8/9
ATTACH,PFNS.
6/7/8/9
```

MFQUEUE COMMAND ALLOWS THE USER TO SEND A TEMPORARY FILE TO THE INPUT OR OUTPUT QUEUE OF THE REMOTE HOST. THE MFQUEUE COMMAND FUNCTIONS MOST LIKE THE ROUTE COMMAND. IN FACT, ROUTE STATEMENT IS PREFERRED TO CDC MAINFRAMES AND FOR MOST FILE TRANSFER BETWEEN NOS AND NON-CDC MAINFRAMES. THE MFQUEUE COMMAND PROVIDES THE CAPABILITY TO SEND A ROUTING DIRECTIVE WITH THE FILE. ONLY WHEN THE USER NEEDS SOME SPECIAL CAPABILITY PROVIDED BY THE ROUTING DIRECTIVE ACCEPTED BY THE REMOTE HOST DOES THE USER REQUIRE MFQUEUE STATEMENT.

EXAMPLE OF ROUTINE FILE TO A REMOTE HOST.

ASSUME THE USER TO BE A MAINFRAME MFA. TO ROUTE LOCAL FILE RBPRINT FROM MAINFRAME MFA TO A REMOTE BATCH PRINTER WITH TERMINAL IDENTIFIER OF AB ON NOS MAINFRAME MFB, AT MAINFRAME MFA ENTER STATEMENTS SIMILAR TO THE FOLLOWING.

```
MFQUEUE(RBPRINT,ST=MFB)
*ROUTE(,DC=PR,TID=AB)
```

THE SAME TASK COULD BE ACCOMPLISHED WITH THE FOLLOWING ROUTE STATEMENT.

```
ROUTE(RBPRINT,DC=PR,TID=AB,ST=MFB)
```

# LESSON 11

# QUEUE FILE CONTROL

## Queue File Control Statements

Recover

QGET

DROP

ONSW  OFFSW

CFO.

GO.  PAUSE.

## OWNERSHIP

THE OWNER OF A QUEUED FILE # IS ALLOWED TO STATUS THE FILE, ATTACH THE FILE, OR DROP THE FILE.

OWNERSHIP OF A FILE IS DETERMINED BY:

1. IF AN INPUT FILE IS BEING QUEUED BY A SUBSYSTEM (SUCH AS BATCHIO OR RBF) THE OWNER OF THE FILE IS DETERMINED BY THE FIRST USER STATEMENT IN THE JOB'S CONTROL STATEMENT RECORD.

2. FOR ANY OTHER TYPE OF QUEUED FILE, THE OWNER OF THE FILE BEING QUEUED IS THE OWNER OF THE JOB DOING THE QUEUEING.

IF USER AAAA ENTERS A ROUTE OR SUBMIT STATEMENT WHICH CREATES A NEW JOB HAVING A DIFFERENT USER NUMBER OF BBBB, USER AAAA IS THE OWNER OF THE NEW JOB AND ALL SUBSEQUENT OUTPUT OR JOBS QUEUED BY THAT JOB.

# NOTE: A QUEUED FILE IS AN INPUT FILE, PRINT FILE, PUNCH FILE, EXECUTING JOB OR DETACHED JOB WHICH HAS A UNIQUE JSN (JOB SEQUENCE NUMBER).

EXAMPLE:

A: ASSUME USER CPS4423 IS LOGON AND ROUTES FILE JOHN TO THE INPUT QUEUE WITH A DIFFERENT USER NAME. THE OUTPUT FILE AND FILE ROUTED BY JOB ABQW BOTH ARE OWNED BY CPS4423.

```
/ROUTE,JOHN,DC=TO.

ROUTE COMPLETE.  JSN IS ABQW.

/ENQUIRE,JSN.


JSN SC CS LID STATUS          JSN SC CS LID STATUS

ABQO.T.ON.      .EXECUTING    ABQX.B.  .  .WAIT QUEUE
ABQW.B.  .  .WAIT QUEUE
```

WHERE THE CONTENTS OF FILE JOHN ARE:

```
ABC.

USER(LAC2172,)

LIMITS,L=JO.

ROUTES,JO,DC=WT,IJN=MARY.
```

A DETACH JOB IS AN INTERACTIVE TERMINAL JOB WHICH IS NO LONGER CONNECTED TO A TERMINAL. DETACHING A JOB IS AVAILABLE ONLY TO INTERACTIVE USERS.

A JOB CAN BE DETACHED AT ANY TIME DURING A SESSION USING ESC-D (HITTING D WHILE ESCAPE KEY IS DEPRESSED) OR CONTROL D DEPENDING ON TERMINAL CLASS. ANY COMMANDS WHICH HAVE BEEN ENTERED AS PART OF A TYPEHEAD QUEUE IMMEDIATELY BEFORE OR AFTER THE DETACH WILL BE RETAINED AS THE JOB STREAMS FOR THE NEW JOB (NOT THE DETACHED JOB).

A DETACH JOB WILL CONTINUE TO EXECUTE IN THE SYSTEM UNTIL EITHER:

A) END OF CONTROL STATEMENTS REACHED.
B) TERMINAL I/O IS ATTEMPTED.
C) TIME LIMIT OR SRU LIMIT IS REACHED.

AT THAT POINT, THE SYSTEM WILL DISPOSE OF THE JOB AS SPECIFIED IN THE MOST RECENT SETJOB STATEMENT. THE JOB WILL EITHER BE SUSPENDED FOR THE DEFAULT TIME PERIOD, OR BE TERMINATED AND ALL FILES DISPOSED TO THE SPECIFIED QUEUE. WHEN A USER DETACHES, THE SYSTEM WILL RELOG HIM IN WITH A NEW JSN NUMBER.

IF SYSTEM OR IAF BECOMES UNAVAILABLE, OR USER BECOMES DISCONNECTED, THE JOB WILL BE AUTOMATICALLY DETACHED.

A USER MAY RECOVER (CONTINUE) A DETACHED JOB WHICH HE OWNS BY USING THE RECOVER STATEMENT.

EXAMPLE 1

```
    .
    .
    .

/COMPASS,I=A,S=NOSTEXT,l=X.
ESC-D                          (TO DETACH)
JOB DETACHED, JSN=ALAB
JSN:ALAC,NAMIAF

RECOVERABLE JOB(S)

JSN      UJN      STATUS      TIMEOUT

ALAB     JOB6     EXECUTING
AANC     JOB3     SUSPENDED   6MIN.

ENTER GO TO CONTINUE CURRENT JOB,
      RELIST TO LIST RECOVERABLE JOBS,
      OR DESIRED JSN:  AANC
JSN:  AANC SYSTEM BATCH SRU:  10.074
STATUS:
CHARACTER SET:  NORMAL MODES:  PROMPT ON
IDLE.  ENTER COMMAND.
```

QGET STATEMENT ALLOWS THE OWNER OF A QUEUED OUTPUT FILE (PRINT, PUNCH, PLOT, OR WAIT FILE) TO MAKE THE FILE A LOCAL FILE ASSIGNED TO HIS CURRENT JOB.

THE LOCAL FILE NAME OF THE ATTACHED QUEUE FILE WILL BE THE JSN VALUE IF THE PARAMETERS UJN OR LFN (LOCAL FILE NAME) ARE NOT SPECIFIED. IF UJN IS SPECIFIED AND LFN IS NOT, THE ATTACH FILE SHALL BE THE UJN VALUE. IF THE LFN IS SPECIFIED, THE LOCAL FILE NAME WILL BE THE LFN VALUE WHETHER OR NOT JSN OR UJN IS SPECIFIED.

```
ENQUIRE, JSN

JSN SC LID STATUS                    JSN SC LID STATUS

ACLU.T.      .EXECUTING        AAVZ.T. .PRINT QUEUE
ACQS.B.      .PRINT QUEUE      ACPN.B. .PRINT QUEUE
ACQU.B.      .PRINT QUEUE

QGET,JSN=ACQS,DC=PR,LFN=JOE
QGET,COMPLETE.
```

ABOVE, LIST ALL THE QUEUED FILES OWNED BY THE USER AND MAKE THE PRINT FILE ACQS A LOCAL FILE CALLED JOE.

WHEN USING UJN, IT MUST BE UNIQUE.

THE DROP STATEMENT ALLOWS THE OWNER OF A QUEUED FILE (OUTPUT FILE OR EXECUTING JOB) TO REMOVE/TERMINATE THE FILE.

A USER MAY NOT TERMINATE HIS CURRENT JOB.

Ex:

```
DROP,UJN=AAVY,DC=ALL
```

WILL DROP QUEUED FILES OWNED BY THE USER AND WHOSE UJN IS AAVY IN ALL THE QUEUES (OUTPUT, WAITING OR EXECUTING QUEUES).

EXAMPLE:

```
ROUTE,GARY,DC=PR.
ROUTE COMPLETE.  JSN IS ACPS.
SETJOB,UJN=MARY,DC=TO.
ROUTE,JOHN,DC=PR,FC=AA,UJN=CPUMT.
ROUTE COMPLETE.  JSN IS ACPT
ROUTE,BILL,DC=PR.
ROUTE COMPLETE.  JSN IS ACPW
ROUTE,CATH,DC=PR
ROUTE COMPLETE.  JSN IS ACPX.
```

ASSUME ABOUT ENVIRONMENT FOR THE FOLLOWING EXAMPLES:

A:

```
QGET,JSN=ACPT,LFN=JOE,DC=PR
QGET,ACPT,PR,,JOE.
QGET,UJN=CPUMT,LFN=JOE,DC=PR.
```

THE THREE COMMANDS ARE EQUIVALENT.
THEY WOULD RETRIEVE THE FILE JOHN AND MAKE IT A LOCAL FILE CALLED JOE.

B:

```
QGET,UJN=MARY,DC=PR.
```

WOULD CAUSE AN ERROR BECAUSE UJN OF MARY IS NOT UNIQUE.

C:

```
DROP,UJN=MARY,DC=PR.
```

WOULD DELETE FILES BILL AND CATH.

THE RECOVER STATEMENT RECONNECTS A USER OWNED DETACHED JOB TO THE TERMINAL. THE CURRENT JOB CONNECTED TO THE TERMINAL IS TERMINATED.

THE USER CAN RECOVER A SPECIFIC JOB WITH

```
RECOVER,JSN=ALAB.
```

OR REQUEST RECOVER DIALOGUE WITH

```
RECOVER.
```

RECOVER DIALOGUE IS INITIATED DURING TERMINAL LOG ON IF THERE ARE RECOVERABLE JOBS.

THE FOLLOWING COMMANDS SET OR CLEAR BITS IN THE FIELD LENGTH OF A USER'S DETACHED JOB. THE EXECUTING JOB MUST CHECK THE STATUS OF BITS TO EFFECTIVELY USE THESE COMMANDS.

ONSW
SWITCH
OFFSW
CFO
GO
PAUSE

# EVALUATION FORM

Course/Seminar Name _____ Date of Attendance From _____ To _____

Instructor _____ Location _____

Please place a rating in the box for each area and then add comments explaining your rating.

*Rating Key*

| | |
|---|---|
| *Excellent* | *5* |
| *Very Good* | *4* |
| *Good* | *3* |
| *Fair* | *2* |
| *Poor* | *1* |

**The Course/Seminar**

* How well did the course/seminar cover the stated objectives? ☐
* To what degree will the course/seminar be helpful in improving on-the-job performance? ☐
* To what extent were the handout materials and visuals helpful in aiding your understanding of the topic? ☐
* What is your overall rating of the organization and content of the course/seminar? ☐

**The Instructor**

* How do you rate the instructor's knowledge of the material and ability to answer questions? ☐
* How effective was the instructor in presenting the material in an understandable manner? ☐
* How effective was the instructor in generating and sustaining interest in the course/seminar? ☐
* How do you rate the instructor's responsiveness to the needs of participants? ☐
* What is your overall rating of the instructor? ☐

**The Facilities**

* How do you rate the appropriateness of the facilities to the topic and means of presentation? ☐
* To what extent were the facilities comfortable, well-lighted and heated or cooled? ☐
* How convenient was the location of the facility? ☐

**EVALUATION FORM**

**Page 2**

**General Comments**

* What changes in the course/seminar would you make if you were the instructor?

* Would you recommend this course/seminar to others in your company or department? Why?

* Please list colleagues or associates who should receive advance notices of similar courses/seminars.

1) Name _____
Organization _____
Address _____
_____
Bus. Tel. No. _____

2) Name _____
Organization _____
Address _____
_____
Bus. Tel. No. _____

3) Name _____
Organization _____
Address _____
_____
Bus. Tel. No. _____

4) Name _____
Organization _____
Address _____
_____
Bus. Tel. No. _____

* Should this course be offered at your company site? If so, who should be contacted to manage it?

* If we may use your comments in future descriptions of the course/seminar, please sign below.

Signature _____
(Optional)

# PARTICIPANT INFORMATION FORM

In order for our seminars/courses to be most effective, they need to take into account the characteristics, needs and objectives of the people who attend them. The information asked for below will assist us in keeping our presentations relevant to the participants and in developing and scheduling new presentations that will meet participant needs. Please complete this form and leave it with the presenter at the next break.

**Seminar/Course Title** _____ **Date of Presentation** _____

**Name** _____ **Field or Type of Business** _____

**Title** _____ **Years of Experience** _____

**Business Address** _____ **Supervisor's Title** _____

_____ **Last professional degree** _____

_____

List your three primary objectives in attending this seminar.

1. _____
2. _____
3. _____

Will this course/seminar be credited toward certification/training requirements? _____

Rank in order of importance in your choice of this seminar session.

**Instructor** _____ **Date** _____ **Location** _____ **Employer's Preference** _____

Previous courses/seminars attended relating to this topic.

1. _____
2. _____
3. _____

Topics for additional courses/seminars in which you would be interested.

1. _____
2. _____
3. _____

What trade journals/magazines do you regularly read or subscribe to in order to keep abreast in your profession?

1. _____
2. _____
3. _____

How did you become aware of this course/seminar?

**Schedule/Catalogue** _____,

**Direct Mail Brochure** _____,

**Recommendations of Supervisor** _____,

**Recommendation of Colleague** _____,

**Corporate Training Department** _____,

**Other** _____ .

# COMMENT SHEET

**MANUAL TITLE:** NOS JOB CONTROL

**PUBLICATION NO.:** FH3020-1 **REVISION:** E

**NAME:** _____

**COMPANY:** _____

**STREET ADDRESS:** _____

**CITY:** _____ **STATE:** _____ **ZIP CODE:** _____

**This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).**

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

**CONTROL DATA CORPORATION**

National Coordinator
Bloomington Facility (MNA02B)
5001 West 80th Street
Bloomington, Minnesota 55437

CUT ALONG LINE

FOLD

FOLD